המחשב האישי למשתמש המקצועי

שליטה בחומרה ובתוכנה של המערכת

משה קליג'
עידו שרון



למחשבי IBN-PC/XT/AT/386 ותואמים
ומחשבי IBM PS/2, PS/1
דיסקט תכניות דוגמה

עורך יצחק עמיהוד

ספרי לימוד והכשרה במדעי המחשב
הוצאת עוד עמי

המחשב האישי
למשתמש המקצועי

להדר ולמשה
מתנה מהמחבר....
יצ
3/1992

מוקדש

לאלו שהתלוננו על הזמן שהקדשנו לכתיבת הספר

עריכה לשונית ועיצוב: שרה עמיהוד

המחשב האישי למשתמש המקצועי

שליטה בחומרה ובתוכנה של המערכת

משה קליג'
עידו שרון

למחשבי IBM\PC\XT\AT ותואמים
ומחשבי IBM PS\2, PS\1
ובנוסף דיסקט תכניות דוגמא

ערך יצחק עמיהוד

ספרי לימוד והכשרה במדעי המחשב
הוצאת עוד עמי

The PC for the Professional User

by

M. Kalige and I. Sharon

edited by I. Amihud

הוצאת הוד-עמי
לספרי מחשבים בע"מ

רח' בילו 3, הרצליה 46426
טל: 052-541207

הודפס בישראל
אלול תשנ"א, ספטמבר 1991

Hod-Ami Ltd
Bilu St. 3, Herzeliya
Israel, September 1991

ISBN 965-361-017-1 מסת"ב

תוכן העניינים

פרק 5 - מערכת הדיסקים/דיסקטים במחשב האישי 178

פרק 6 - הפעלת תכניות תחת מערכת ההפעלה DOS 231

פרק 7 — וירוסים במחשב האישי 244

הקדמה

המחשב האישי הפך לכלי עבודה על שולחנם של רבים במגזרי תעסוקה ותפקיד שונים: הכתבנית, שלה הוא יועד בתחילה, המינהלן ואיש הכספים, המנהל, התלמיד, העובד במחסן המפעל או ברצפת הייצור, המדען ו..., ומי־לא?

בתחילה השתמשו במחשב האישי למשימות כמו אלו של מכונת כתיבה, או מכונת חישוב משוכללת, או שילוב של שתיהן. לאחר זמן הוחל לנצל את התכונות והאפשרויות הייחודיות הגלומות בו ולהוסיף לו פריטי חומרה ותוכנה כדי להפיק ממנו יותר. כך אנו מוצאים, שבמקום מחשב אישי פשוט ותמים נמצא לפנינו מחשב של ממש, בעל חיוניות הולכת וגוברת כתוצאה מתוספות ושיפורים שעורכים בו ועם היצע נרחב של כישורים חדשים.

כל אלה מעמידים את המשתמש ההדיוט במצב מביך, מכיון שהוא מחזיק ברשותו מערכת מחשב עתירת אפשרויות, אשר רק חלק מהן באות לידי ביטוי כתוצאה מחוסר היכרות מעמיקה שלו עם אפשרויות אלו. פן אחר של הבעיה היא ההכרה שאפשר לרתום את המחשב האישי למשימות אשר חורגות משימושי המינהל, אולם לצורך זה יש לחדור ל"תחום הפרט" שלו. כדי לנצל כישורים מיוחדים אלה של המחשב יש להגיע אל הרכיבים הפנימיים, או לפחות אל הכניסה והמוצא שלהם. צריך לעתים להוסיף רכיבים והתקנים לביצוע מטלות מיוחדות, כמו קלט נתונים והפעלה של מיכשור חיצוני. משימות אלו חורגות מיכולתו של המשתמש הרגיל העוסק בתוצאות השימוש ולא בדרך הביצוע של המטלות.

הספר שלפנינו מכוון אל המשתמש המקצועי במחשב האישי (PC) של יבמ ותואמיו, או במערכות PS/2 ו-PS/1 של יבמ. הוא מכוון אל מי שמעוניין להכיר את רכיבי החומרה והתוכנה של המחשב האישי, כדי שיוכל לבחון ולהעריך את הדרכים להפיק את המירב מהמחשב, להעלות את ביצועיו ולהוסיף עליהם, לבחון תקלות אפשריות ולהתגבר עליהן כאשר הן קורות.

הספר מציג את עקרונות הפעולה, התכונות והרכיבים של משפחת מעבדי אינטל 80X86 והמעבדים 8088/86 ומסביר בהרחבה את ההבדלים בין המחשבים האישיים השונים. הקורא ילמד על מרכיבי החומרה של הלוח הראשי, מערכת ה-BIOS, סוגי זיכרון, כרטיסי התיאום להתקנים חיצוניים והתקני קלט/פלט השונים, תפקידיהם והשילוב ביניהם לבין מערכת ההפעלה DOS.

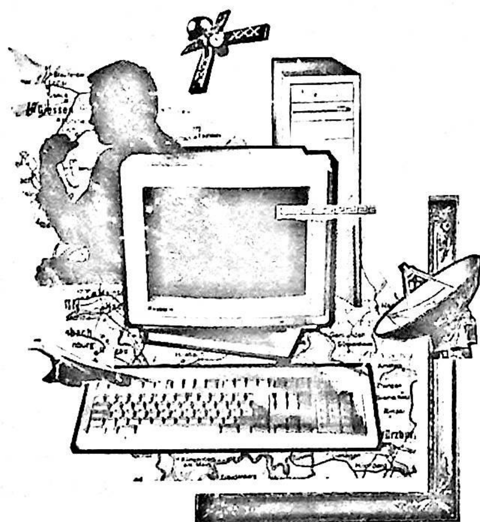
הביטוי של המחשב בעיני המשתמש הם התקני הקלט/פלט, כמו המקלדת, המסכים לסוגיהם, המדפסות, התקשורת האסינכרונית וכמובן – הדיסקים. המחברים מציגים בהרחבה את הדיסקטים והדיסקים: אופני השימוש, כיצד לשפר ביצועים ולתחזק וכיצד להגן על הנתונים וכיצד לשחזר ממחיקה בשוגג. המחשב האישי מתקשר לסביבתו שבה מחשבים אישיים ומחשבים מרכזיים באמצעות רשתות תקשורת מקומיות ורחבות, אשר להן הוקדש פרק נפרד.

החומרה ומערכת ההפעלה הן המצע לביצוע תכניות היישום. למבנה קובצי שפת מכונה ולדרך שבה מערכת ההפעלה מפעילה את תכנית היישום ניתנה תשומת לב רבה. נושא מרכזי בספר הוא הקשר שבין הרכיבים השונים במערכת המחשב ובינם לבין התוכנה, אשר מתקיים באמצעות מערכת הפסיקות (Interrupts) של החומרה ושל מערכת ההפעלה DOS.

המחברים מכוונים ומנחים את המשתמש לנצל ביעילות את עוצמת פעולתה של מערכת המחשב שברשותו. לשם כך, המשתמש מקבל הסברים והדגמה של פקודות DOS אשר רלוונטיות לנושאי הדיון השונים ופקודות אשר דרושות לניהול ותחזוקה של המערכת ושל תכניות היישום. המחברים מסבירים את הפקודות החדשות של DOS 5, אשר רשימה מפורטת שלהן ניתנת בנספח. הם מלווים את הספר בתכניות דוגמה רבות, אשר מוצגות בנספח וניתנות במלואן בתקליטון המצורף.

וירוסים הם רעה חולה שעלינו להתמודד אתה ועל כן המחברים מסבירים את המבנה שלהם ודרך פעולתם, מדריכים כיצד להתגונן מפניהם ואף מציגים תכנית וירוס ידידותית לשם הדגמה.

אנו מקווים שהלימוד בספר והתרגול באמצעות תוכניות הדוגמה המצורפות יאפשרו לך לנצל באופן יעיל יותר את מערכת המחשב שעומדת לרשותך ולפתח באמצעותה יישומים מתקדמים. תוכל להיעזר בספר גם אם אתה נמצא "על פרשת דרכים" ומתלבט באיזו מערכת חדשה לבחור, איך להרחיב את אפשרויות המערכת שלך, או כיצד לאזן את הביצועים שלה.



פרק 1

רענון עקרונות פעולה של מחשבי משפחת 86

פרק זה מיועד לרענון הידע במבנה המחשב ועקרונות פעולתו. הנושאים ערוכים מן הקל אל הכבד, בדרך אשר תאפשר לקורא הבקיא בחומר לדלג על נושאים אשר מוכרים לו.

1.1 מבוא

עקרונות העבודה של מערכת המחשב האישי מוליכים אותנו לשיטות עבודה, שונות בתכלית מהדרך שבה אנו חושבים ומתפקדים בחיי היומיום. צורת עבודה שונה זו ומהירות הביצוע של המחשב הובילו למיתוסים רבים הקשורים במחשב. גם אנשים הקשורים לעולם המחשבים רואים לפעמים את המחשב כ"מכשף" ולא כ"מחשב". כוונת הספר לפזר את חוסר הבהירות, לרענן מושגים מוכרים ולעשות היכרות עם נושאים טכניים חדשים הקשורים למחשב האישי.

מערכת המחשב נועדה לאפשר למשתמש להזין **נתוני קלט**, לאחסן אותם לזמן קצר או ארוך ולבצע תהליך **עיבוד נתונים** על פי תכנית אשר נכתבה מראש בידי מתכנת. בסוף תהליך העיבוד **מופק פלט** – תוצאת העיבוד – לשם תמיכה בפעולות התפקודיות של המשתמש בארגון, או בחייו הפרטיים.

1.2 ייצוג הנתונים

שיטת ייצוג הנתונים במחשב שונה משיטת ייצוג הנתונים שאנו מכירים בחיי היומיום. אנו רגילים לחשב ולכתוב מספרים בשיטה העשרונית, לכתוב מכתבים ותמליל בשפה העברית, או בשפה אחרת, ומשתמשים לצורך זה בתמונות תווים המוכרות לכל. המחשב אינו מאחסן את המידע בצורה בה אנו חושבים ועובדים. כאשר קיים קשר ישיר בין המשתמש לבין המחשב, המחשב מקבל מהמשתמש את הקלט מהמקלדת ומוציא לו את הפלט בצורה של הצגה על גבי מסך, או על גבי נייר. נראה הדבר למשתמש, כאילו המחשב משוחח איתו באותה שפה. פעולה זאת היא רק למראית עין, מכיון שתהליכי ההצגה של הנתונים למשתמש נועדו אך ורק לצורך זה, ואינם נדרשים למחשב לצורך ביצוע הפעילות הפנימית. להיפך, פעולות התרגום של הנתונים לשם הצגתם למשתמש רק מאיטות את פעילות המחשב.

הנתונים מיוצגים בתוך רכיבי המחשב **בשיטה הבינארית**, הפועלת לפי בסיס 2, שבו קיימים אך ורק שני צירופים שונים שמהם יוצרים את התבניות של כל הנתונים. כלומר, כל נתון המאוחסן במחשב מתורגם לסדרות של מספרים בינאריים. במחשב קיימים סוגים רבים של התקנים ורכיבים המסוגלים לאחסן נתונים. הם בנויים בטכנולוגיות שונות ומאחסנים את הנתונים בדרכים שונות. אך עם זאת, העיקרון המשותף לכל ההתקנים הללו הינו צורת הייצוג הבינארית (שני מצבים בלבד).

ייצוג בינארי	ייצוג עשרוני	ייצוג הקסה-דצימלי
00000	0	0
00001	1	1
00010	2	2
00011	3	3
00100	4	4
00101	5	5
00110	6	6
00111	7	7
01000	8	8
01001	9	9
01010	10	A
01011	11	B
01100	12	C
01101	13	D
01110	14	E
01111	15	F
10000	16	10
.....	..	..

ייצוג נתונים בשיטות ספירה שונות

הייצוג של הנתונים והמידע המעובד במחשב הינו בשיטה הבינארית, אך מטעמי נוחות של המשתמשים, אין אנו משתמשים בייצוג זה, אלא בצורה הנוחה יותר לשימוש (נסה לקלוט סדרה של 16 סיביות ולהבין את משמעותן). לעתים אנו "מתרגמים" את סדרת הסיביות האלו ל-2 תווים אלפאנומריים, או ל-4 ספרות בשיטה ההקסה-דצימלית, שבה ההצגה נוחה וברורה יותר.

תרגום משיטה בינארית לבסיס 16 (הקסה-דצימלי)

כדי להפוך ייצוג של מספר בשיטה הבינארית לייצוג הקסה-דצימלי, מחלקים את סדרת הסיביות לקבוצות של 4 סיביות (מימין לשמאל). לכל צירוף של 4 סיביות מתאימים ספרה הקסה-דצימלית אחת (ראה טבלה ייצוג) ולכן, סדרת הסיביות "0101 1110 1010 0110" (בת 16 סיביות) תחולק ל-4 ספרות הקסה-דצימליות ותתורגם למספר 5EA6Hex (על פי הטבלה, הערך 0101 בינארי יתורגם ל-5 וכו').

1.3 הגדרות בסיסיות

נחזור על מספר הגדרות:

- **סיבית** (Bit) - יחידת המידע הקטנה ביותר לאחסון מידע, אשר מסוגלת לייצג שני מצבים בלבד: אפס ואחד, שחור ולבן וכו'.

- **בית** (Byte) - סדרה של 8 סיביות, אשר מייצגות בית אחד (לדוגמה אותיות, ספרות, סימני פסוק). כל פעולה המתבצעת במחשב על סימנים אלה מתבצעת בפועל על סדרות הסיביות האלו, ורק כאשר אנו מבקשים מהמחשב להציג את המידע, נעשה "תרגום" לצורת התווים שאנו מכירים (ראה פרק-הצגת תווים במסך ונספח טבלת ASCII לייצוג תווים במחשב).

- **מלה** (Word) - סדרה של 16 סיביות (2 בתים) המייצגות יחידת מידע גדולה יותר.

- **מלה כפולה** (DWord) - סדרה של 32 סיביות (4 בתים).

- **אוגר** - (Register) - התקן אשר יכול לאחסן סדרת סיביות שלמה (בית או מלה). כאשר מאחסנים בתוכו סדרת סיביות, הוא יחזיק את ערכה עד אשר נשנה את תוכנו. הפקודה Write גורמת לכך שהתוכן הקודם של האוגר נמחק ותוכנו יהיה סדרת הסיביות החדשה שנכתבה אליו. אפשר לקרוא מתוך האוגר את סדרת הסיביות הקיימת בו באמצעות הפקודה Read. תוכן האוגר לא משתנה במקרה זה.

- **זיכרון** (Memory) - הזיכרון הינו התקן אלקטרוני המאחסן בתוכו מידע לוגי בצורה ספרתית. המידע יכול להיות הן תכניות והן נתונים. מאפיינים עיקריים לרכיבי זיכרון הינם:

  * **יכולת כתיבה/קריאה או קריאה בלבד** (Read/Write or Read Only). ניתן להבחין במספר סוגים של התקני זיכרון הנבדלים זה מזה בתכונה זאת ובטכנולוגיה שבה הם בנויים.

  * ROM / EPROM - התקני זיכרון המאפשרים פעולת קריאה בלבד. את הנתונים או התכניות האגורים בהתקנים אלו רושמים באמצעות מכשיר חיצוני למערכת המחשב, ורק לאחר שמידע זה קיים בהם, הם מוכנסים למערכת המחשב. המאפיין העיקרי של התקנים אלה הינו יכולת שמירת המידע לזמן ארוך ללא תלות באספקת מתח חיצוני. רכיבים אלה משמשים בעיקר לאחסון תכניות הנדרשות להפעלה ראשונית של המחשב.

  * RAM **סטטי/דינמי** (S-RAM/D-RAM) - התקן זיכרון המאפשר פעולות קריאה וכתיבה. המידע האגור ברכיבי זיכרון אלה נשמר כל עוד ההתקן מחובר למקור מתח. התקנים מסוג זה משמשים לאחסון זמני של נתונים ותכניות. התקנים D-Ram מחייבים מעגל חומרה נוסף שמבצע רענון של הנתונים האגורים בהם. התקני S-Ram אינם מחייבים מעגל זה.

* **רוחב הזיכרון** (Memory Width) - מספר הסיביות שכותבים או קוראים לרכיב הזיכרון בפעולה אחת.

* **גודל הזיכרון** (Memory Size) - מספר הכתובות שקיימות לאחסון נתונים ותכניות. גודל הזיכרון נמדד במספר הבתים בזיכרון.

* **זמן גישה** (Access Time) - משך הזמן הדרוש לזיכרון כדי לקבל, או למסור נתונים כשפונים אליו. זהו נתון חשוב בתכנון סוג הזיכרון הנדרש למערכת. נתון זה קובע אם הזיכרון מסוגל למסור או לקבל את הנתונים מהמעבד הראשי בקצב שבו המעבד פועל, או שהמעבד נדרש לפעול במהירות נמוכה יותר, כאשר הוא ניגש לזיכרון. שימוש בזיכרון, אשר זמן הגישה אליו ארוך יותר מהזמן הנדרש על ידי המעבד המרכזי לביצוע הפעולה, מאיט את קצב המערכת ובכך מקטין את ביצועי המערכת כולה.

▪ **מעבד** – CPU (Central Processing Unit) - התקן המכיל בתוכו רכיבים אחדים:
- אוגרים. משמשים לאחסון ביניים של נתונים, לשמירת מידע על מצב המערכת וניהול הכתובות.
- יחידה אריתמטית/לוגית – ALU (Arithmetic Logic Unit). מאפשרת להעביר נתונים בין האוגרים השונים ולבצע פעולות מתמטיות ולוגיות על נתונים שנמצאים באוגרים ובזיכרון.
- יחידת בקרה. מאפשרת לשלוט על העולם החיצוני, לקרוא נתונים ולרשום נתונים לעולם חיצוני זה. מבנה המעבד מאפשר לו לבצע פעולות עיבוד נתונים ובקרה.

התכונה / סוג זיכרון	ROM / EPROM	RAM סטטי Static RAM	RAM דינמי Dynamic RAM
שמירת נתונים	לא תלוי במתח. הנתונים צרובים בצורה מכנית על גבי ההתקן.	הנתונים נשמרים כל עוד ההתקן מחובר למקור מתח.	דורש רענון שוטף של הנתונים. ללא רענון הנתונים. הנתונים ישתבשו לאחר זמן קצר.
מעגל חומרה	פשוט	פשוט	מורכב. דורש מעגל לצורך רענון.
צורת רישום הנתונים	בית חרושת, או צריבת הזיכרון מחוץ למערכת המחשב.	תוך כדי עבודה של המחשב אפשר לבצע פעולות קריאה/כתיבה.	תוך כדי עבודה של המחשב אפשרי לבצע פעולות קריאה/כתיבה.
מחיר/קיבולת	גבוה יחסית	בינוני	נמוך יחסית
גודל פיסי	גדול יחסית	גדול יחסית	קטן יחסית

טבלת השוואה בין זיכרונות

- **מעבד עזר מתמטי** (Mathematic Co-Processor) - התקן עזר אשר מתחבר כתוספת למעבד ומאפשר לו לבצע פעולות מתמטיות על נתונים ממשיים. ביצוע פעולות אלו בחומרה מאיץ את מהירות העיבוד.

- **מרחב זיכרון** - במושג מרחב הזיכרון נתייחס תמיד למרחב זיכרון RAM. אם לדוגמה, למחשב שלנו יש זיכרון בגודל 640KB, פירוש הדבר שיש בו 655360 בתים, או 5242880 סיביות שבהן אנו יכולים לאגור תכניות או נתונים. המרחב הזה מוגדר בבתים בני 8 סיביות כל אחד, אשר את הראשון נסמן "בית 0" ואת האחרון נסמן "בית 655360".

- **מיפוי זיכרון** - בתהליך זה מחלקים את הזיכרון לקטעים שלכל אחד מהם מוגדר תפקיד, או ייעוד. כל פניה של המעבד לקטע זיכרון זה תשמש לייעוד זה בלבד. לדוגמה, אם תכנית מסוימת נמצאת בזיכרון בתחום הכתובות 2000-3000Hex, נאמר כי היא ממופית לכתובת 2000-3000Hex. כך אפשר למפות טבלה, או איזור עבודה של קלט/פלט בקטע זיכרון.

- **מרחב קלט/פלט** (I/O) - המעבד שולט על שני מרחבי כתובות - מרחב הזיכרון ומרחב הקלט/פלט. במרחב הקלט/פלט ממופים התקני הקלט והפלט של המחשב כדי לאפשר למעבד גישה נוחה אליהם. כלומר, הוא דומה למרחב הזיכרון מבחינה זו שגם ממנו יכול המעבד לקרוא ולכתוב. ההבדל הוא בכך שמרחב זה קטן בהרבה ממרחב הזיכרון ואין בו מקטעים והיסטים (ראה להלן) ולכן הגישה אליו נוחה ומהירה.

- **מקטע** (Segment) - איזור בזיכרון בגודל של 64KByte. בתוך המעבד קיים אוגר מקטע המצביע על כתובת התחלתית של המקטע ומאפשר לו לגשת לכל כתובת במרחב של 64KByte מכתובת הבסיס והלאה, בלי לשנות את ערכו של אוגר המקטע. על מנת לגשת לאיזור הנמצא מחוץ ל-64KByte שעליהם מצביע אוגר המקטע, יש צורך לשנות את תוכנו של האוגר.
- **היסט** (Offset) - הערך שיש להוסיף לערך של "כתובת בסיס" על מנת להגיע לכתובת הפיסית שבה מאוחסן נתון כלשהו. דוגמה: אם נאמר על נתון שהוא מאוחסן ב-Offset של 100Hex, לא נדע את המיקום הפיסי שלו במרחב הכתובות. אך אם נאמר כי מיקומו הוא 100Hex יחסית לכתובת מוחלטת 1000Hex, נוכל לחשב בקלות כי כתובתו הפיסית של הנתון היא 1100Hex (כתובת הבסיס + הכתובת היחסית).

- **הבאת פקודה** (Fetch) - על מנת שהמעבד יוכל לבצע פקודה כלשהי עליו להביא אותה ליחידת הבקרה שבמעבד לצורך פענוח. מחזור המכונה המאפשר הבאה של הפקודה ליחידת הבקרה שבמעבד נקרא Fetch.

- **מיעון ישיר** (Direct addressing) - צורת גישה לזיכרון בפקודה מתוך אוסף הפקודות של המעבד. הכתובת שאליה מתייחסים לצורכי הבאת נתון מזיכרון או כתיבה של נתון לזיכרון מוזכרת בתוך הפקודה עצמה.
לדוגמה, הפקודה MOV AX,COUNT תכניס את תוכן התא COUNT ל-AX כאשר התא COUNT מוגדר כמשתנה ברוחב 2 בתים (כתובת COUNT מופיעה בפקודה).

- **מיעון עקיף** (Indirect addressing) - צורת גישה לזיכרון בפקודה מתוך אוסף פקודות של המעבד. הכתובת שאליה מתייחסים לצורך הבאת נתון מזיכרון או כתיבה של נתון לזיכרון איננה מוזכרת בתוך הפקודה

עצמה. בפקודה מוזכר רק המקום בו נמצאת הכתובת, שאליה יש לרשום או שממנה יש לקרוא את הנתון.
לדוגמה, הפקודה MOV AX,[SI] תכניס לאוגר AX את תוכן של הכתובת שנמצאת באוגר SI. כלומר, הכתובת לא מופיעה בפקודה ויש צורך בתהליך תרגום כדי לחשב את ערכה הנכון.

1.4 משפחת מעבדי אינטל 8088/8086

כדי לענות על השאלה כיצד פועל מחשב PC וכיצד הוא מבצע תכניות, עלינו להסתכל על המעבד (CPU) המהווה את לב מערכת המחשב ומקנה לה את כושר הביצוע. המעבד שייך למשפחת מעבד INTEL iAPx86. בדרך כלל נמצא במערכת המחשבים האישיים את המעבד 8088, אך קיימות מערכות מחשבים אישיים המבוססות על מעבדי 8086, 80286, 80386 או 80486. המעבדים מסדרת 80X86 הם המתקדמים יותר. זוהי "משפחת מעבדי 86".

כדי להסביר על מבנה המחשב נבחן את המבנה של מעבד 8088 אשר מותקן במחשבי יבמ מדגמי PC/XT ותואמיהם. כדי לשמור על תאימות של המערכות השונות ולהשתמש בתכונות מערכת ההפעלה DOS, פעלו היצרנים כדי שגם אם יימצא מעבד אחר בתוך המערכת, הוא יפעל בדרך דומה ככל האפשר למעבד 8088. את ההבדלים בין המעבד 8088 לבין מעבדים בעלי ארכיטקטורה מתקדמת יותר נפרט בהמשך.

הארכיטקטורה של מעבדים אלה הינה ארכיטקטורת צינור (Pipe-line), אשר כוללת שתי יחידות נפרדות בתוך המעבד. **יחידת העיבוד** EU (Execution Unit) אחראית על **ביצוע ההוראות**, ואילו **יחידת הקישור** BIU (Bus Interface Unit) אחראית על התקשורת בין המעבד לעולם החיצוני, כגון **הבאת הוראות** (Fetch), **קריאת נתונים וכתיבת תוצאות**. יחידות אלו יכולות לפעול בצורה בלתי תלויה ובמקביל, בהסתמך על **עיקרון המקומיות** (Locality) בתכניות האומר שבאחוז גבוה של המקרים ההוראה הבאה שתתבצע תהא ההוראה הבאה בתור בזיכרון. זמן הבאת הוראה חדשה כמעט ולא קיים, כי במרבית המקרים הוראה זו כבר תימצא מוכנה ב-BIU בגמר ביצוע הוראה קודמת על ידי EU. בכך משפרת ארכיטקטורה זו את ביצועי המעבד. למעבדים 8088 ו-8086 ניתן לחבר מעבד עזר מתמטי 8087.

1.4.1 מעבדים אחרים

בעולם המחשבים קיימות משפחות נוספות למשפחת מעבדי "86". הארכיטקטורה של מחשבים קטנים מתחלקת לשתי קבוצות:

* CISC (Compound Instruction Set Computer) - מעבדים בעלי אוסף פקודות מכונה גדול יחסית. ביצוע של כל פקודת מכונה נמשך מספר מחזורי "שעון" בהתאם למורכבות הפקודה.
* RISC (Reduced Instruction Set Computer) - מעבדים בעלי אוסף פקודות קטן. משך הביצוע של כל פקודה הוא מחזור "שעון" אחד. כלומר, זמן ביצוע פקודה הוא קטן מאוד וכל הפקודות השונות מתבצעות בזמן אחיד.

משפחת 86 היא משפחת מעבדים אחת מתוך רבות בקבוצת CISC. משפחה חשובה אחרת היא משפחת 68000 של Motorola, שמשתמשים בה במחשבי מקינטוש (Macintosh). מחשב אישי שהמעבד שלו אינו ממשפחת 86, בהכרח אינו תואם יבמ (אין זאת אומרת שכל מחשב עם מעבד 86 הוא תואם). מעתה נעסוק רק במשפחת מעבדי 86 הקיימת במחשבים תואמי יבמ.

1.5 האוגרים במעבד

את האוגרים במעבד אפשר לקבץ לפי תפקידיהם. נסקור תחילה את קבוצות האוגרים השונות ואחר כך נפרט את תפקידיהם. כל האוגרים במעבד הינם בגודל של 16 סיביות.

אוגרים השייכים ל-EU

AX	AH	AL	Accumulator	צובר
BX	BH	BL	Base	אוגר בסיס
CX	CH	CL	Count	מונה
DX	DH	DL	Data	נתונים

General Purpose Registers - אוגרים לשימוש כללי

SP	Stack Pointer	מצביע מחסנית
BP	Base register	מצביע בסיס
SI	Source Register	מצביע מקור
DI	Destination Register	מצביע יעד
FLAGS	Flags Register	אוגר דגלים

Special Purpose Registers - אוגרים ייעודים

אוגרים השייכים ל-BIU

CS	Code Segment	מצביע מקטע תכנית
DS	Data Segment	מצביע מקטע נתונים
SS	Stack Segment	מצביע מקטע מחסנית
ES	Extra Segment	מצביע מקטע נוסף
IP	Instruction Pointer	מצביע הוראות

אוגרי מקטע (מצביעי כתובת) - Segment Registers

1.5.1 אוגרים לשימוש כללי

אוגרים אלה משמשים בדרך כלל לאחסון ביניים של נתונים שונים לצורך ביצוע פעולות במעבד. יש פעולות שניתן לבצע על ידי אוגרים מסוימים בלבד, אך יש פעולות רבות של המעבד שניתן לבצע בכל אחד מהאוגרים במידה שווה.

דוגמאות לפעולות ייחודיות:

AX - באוגר זה מבצעים פעולות מתמטיות, כמו חילוק וכפל, פעולות קלט/פלט, פעולות בערכים דצימליים (BCD) וכו'.

BX - האוגר מכיל כתובת בסיס לצורך פנייה לאיזור זיכרון כלשהו. במקרה זה, ההתייחסות לכתובת הזיכרון הינה על ידי הוספת ערך היסט (Offset value).

CX - האוגר משמש כמונה אוטומטי בביצוע פעולות מסוימות של המעבד, כגון פקודת המכונה: MOVS, LOOP וכו'.

DX - זהו אוגר כתובת בפעולות ק/פ (I/O). הוא מחזיק נתונים בפעולות מסוימות של המעבד, כגון כפל וחילוק.

באוגרים AX, BX, CX, DX ניתן להשתמש כאוגר יחיד ברוחב של 16 סיביות, או כשני אוגרים נפרדים כשכל אחד מהם הינו ברוחב של 8 סיביות. במקרה זה נעשה שימוש בשמות הנוספים xL, xH. לדוגמה: האוגר AX הינו ברוחב של 16 סיביות וניתן לגשת אליו גם באמצעות השמות AL, או AH המציינים אוגרים ברוחב של 8 סיביות.

1.5.2 אוגרים ייעודיים

SP - האוגר מצביע לראש המחסנית ומשמש **כערך היסט** (offset) למקטע המחסנית SS בלבד. לא ניתן לגשת באמצעותו לאוגר מקטע אחר, ולא ניתן להשתמש בו למטרה אחרת.

BP - מצביע מחסנית נוסף, אשר משמש בעיקר כאינדקס למחסנית לצורך העברת פרמטרים בין תכניות. אוגר זה מהווה בדרך כלל כתובת היסט לאוגר המקטע SS. אם קיים צורך להשתמש בו ככתובת היסט לאוגר מקטע אחר ניתן לעשות זאת על ידי ציון מפורש של אוגר המקטע שאליו יש להתייחס.

DI, SI - אוגרים אלה משמשים כאוגרי אינדקס לגישה לתכניות ונתונים בשיטת מיעון עקיף, בנוסף לאפשרות השימוש בהם כאוגרים כלליים.

IP - Instruction Pointer - מצביע הוראות. משמש כמצביע לכתובת היסט של הוראה בתוך מקטע הקוד. הוא פועל אך ורק ככתובת היסט ל-CS.

1.5.3 אוגר דגלים

אוגר הדגלים (Flag register) מכיל "דגלים" לשימוש התכנית. דגלים אלה מושפעים מביצוע פעולות במעבד. על כן, התכנית יכולה לבדוק את מצבם ולהגיב למצבים משתנים על ידי ביצוע קטע תכנית שונה בהתאם למצב של כל אחד מהדגלים שבאוגר.

הגדרה: **פעולה קודמת**, לגבי אוגר הדגלים, היא פעולת מעבד אשר משנה את מצב הדגלים. לא כל הוראה משנה את מצב הדגלים והוראה המשנה את מצב הדגלים אינה משנה בהכרח את כולם.

מבנה אוגר הדגלים:

סיביות	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	CF		PF		AF		ZF	SF	TF	IF	DF	OF				

CF - Carry Flag - דגל הנשא. ערכו "1" כאשר בפעולה החיבור שהתבצעה היה **נשא** (Carry), או שבפעולת החיסור היה **לווה** (Borrow).

PF - Parity Flag - דגל הזוגיות. ערכו של דגל זה "1" כאשר מספר הסיביות המכילות "1" בתוצאה יהיה זוגי, ואם לא - ערכו יהיה "0".

AF - Auxiliary Flag - דגל נשא עזר. הדגל מגיב כמו דגל הנשא, אך מתייחס למצב שבו יש נשא או לווה בין סיבית 3 ל-4 בפעולה. משמש בעיקר בפעולות BCD.

ZF - Zero Flag - דגל האפס. ערכו "1" כאשר תוצאת החישוב היא אפס, ואם לא - ערכו יהיה "0".

SF - Sign Flag - דגל סימן. ערכו "1" כאשר סיבית MSB בפעולה קודמת היה "1". הוא משמש לציון מספרים שליליים וחיוביים בשיטת המשלים ל-2.

TF - Trap Flag - דגל מלכודת. ערכו "1" כאשר נמצאים בצורת עבודה של צעד יחיד, שבה מבצעים כל פקודה בנפרד ולא ברצף. הדבר שימושי בעיקר בכלים לניפוי שגיאות תוכנה (Debuggers). הדגל מציין למעבד להפסיק ביצוע בגמר כל פקודה ולהעביר שליטה לתכנית פסיקת Single Step.

IF - Interrupt Enable Flag - דגל הפסיקה. כשדגל זה מופעל הוא מתיר פסיקות במערכת. הוא ניתן לשליטה בתוכנה בפקודות Cli ו-Sti.

DF - Direction Flag - דגל כיוון. משתמשים בדגל זה בפעולות על מחרוזות. ערכו של דגל זה קובע את כיוון ההתקדמות בכתובות הזיכרון שעליהן מתבצעת הפעולה (לפנים או לאחור).

OF - Overflow Flag - דגל גלישה. ערכו "1" כאשר תוצאת הפעולה חורגת מגודל האופרנד שעליו התבצעה הפעולה. הוא מציין כי התוצאה אינה מדויקת, או אינה נכונה כלל. יש לבדוק גם את המצב של OF כדי לדעת אם תוצאה הפעולה האריתמטית אמנם נכונה.

1.5.4 אוגרי מקטע

אוגרי המקטע מכילים את הכתובת ההתחלתית של כל מקטע ייעודי במרחב הזיכרון של המעבד.

- CS - האוגר מצביע לכתובת התחלתית של מקטע התכנית. המעבד משתמש במקטע זה בצורה אוטומטית בכל פעם ש"מביאים" פקודה מהתכנית.
- DS - האוגר מצביע לכתובת התחלתית של מקטע הנתונים שעליהם התכנית פועלת.
- SS - מצביע למחסנית המשמשת לאחסון נתוני ביניים וכתובות לשימוש עתידי בתכנית. כל פעולות POP, PUSH, CALL וגישות דרך אוגר BP מתבצעות על אוגר זה.
- ES - מצביע למקטע נתונים נוסף.

1.6 הערוצים (Buses) השונים במערכת

המעבד הינו לב המערכת, אך כמו לב אנושי, הוא אינו יכול לפעול כמערכת ללא ההתקנים ההיקפיים המאפשרים לו לתפקד. התקשורת בין המעבד להתקנים אלה נעשית ביוזמת המעבד ובאמצעות 3 ערוצים קשר עיקריים:

<u>הגדרה:</u>

ערוץ – Bus – הינו קבוצת חוטים בעלת יעוד משותף, המעבירים נתונים בערכים של 0 ו-1:

* **ערוץ כתובות** – Address Bus: על מנת לאפשר למעבד להתקשר אל התקנים היקפיים המחוברים אליו, עליו להודיע בכל פעולה שהוא יוזם למי מהם מיועדת הפעולה המתבצעת. בחירת ההתקן נעשית באמצעות כתובת הניתנת לכל התקן ולכן, כאשר המעבד רוצה להתקשר להתקן, עליו להציג על ערוץ הכתובות את הכתובת המתאימה. ההתקן מצידו חייב "להאזין" כל הזמן לכתובות הנמצאות על ערוץ הכתובות על מנת לוודא מתי פונה אליו המעבד. רוחב ערוץ הכתובות של המעבד 8088 הינו 20 סיביות המאפשרות להגדיר עד מיליון כתובות שונות במערכת (2^{20} צירופים שונים).

* **ערוץ נתונים** – Data Bus: דרך ערוץ הנתונים שולח המעבד נתונים (Data) להתקנים ההיקפים המחוברים אליו, או שהוא מקבל מהם נתונים. המעבד יעביר או יקבל נתונים אל ההתקן באמצעות כתובת מתאימה בערוץ הכתובות. במעבד 8088 רוחב ערוץ הנתונים הוא 8 סיביות ואילו במעבדי 8086 ו-80286 רוחבו 16 סיביות. ערוץ נתונים ברוחב 16 סיביות מאפשר להעביר בפעולה יחידה, פי 2 נתונים מערוץ הנתונים של מעבד 8088 שרוחבו 8 סיביות.

* **ערוץ הבקרה** – Control Bus: באמצעות ערוץ הבקרה המעבד מודיע על סוג הפעולה המתבצעת. באמצעותו מתאם המעבד את זמני הביצוע בין ההתקנים השונים (מתי לקבל את הנתונים דרך ערוץ הנתונים, מתי להוציא נתונים, האם זו פעולת קריאה או כתיבה וכו').

הערה: בצורת ההתקשרות והחיבור של המעבד להתקני קלט/פלט השונים נדון בהמשך בפרקים העוסקים בהתקנים השונים.

1.7 גישה למרחב הכתובות של המעבד

המעבד מספק ערוץ כתובות (Address Bus) של 20 סיביות ובכך הוא מאפשר גישה למרחב זיכרון של 1 מגה בתים (2^{20}). זה הוא תחום כתובות הזיכרון 00000Hex-0FFFFFHex. המעבד רואה את הזיכרון כרצף אחד (Bank) של מיליון בתים, אשר לכל בית ברצף זה ישנה הכתובת המגדירה אותו. במעבדים 8086, 80286 הזיכרון מחולק פיסית לשני חלקים שונים. חלק אחד מכיל את כל הכתובות הזוגיות וחלק שני מכיל את כל הכתובות האי-זוגיות. המעבד יכול להתייחס לזיכרון זה כזיכרון של מלים ברוחב של שני בתים, או כזיכרון רציף ברוחב של בית אחד. הדבר אפשרי מכיון שאוסף הפקודות של המעבד מאפשר לגשת לפי הצורך לבית אחד או למלה.

התכניות המתבצעות במעבד רואות את הזיכרון בצורה לוגית כמחולק למקטעים בני 64KByte. עובדה זו נובעת ישירות מהמבנה הפנימי של המעבד, שבו יש אוגר מקטע ברוחב של 16 סיביות המאפשר גישה ל-64KByte. כל מקטע כזה יכול להימצא בכל כתובת שהיא במרחב הזיכרון, ובתנאי שכתובת זו מהווה כפולה של 16 (10 בספירה הקסה-דצימלית). כל כתובת במרחב הזיכרון מחושבת באופן יחסי לאוגר המקטע בהתאם לסוג הפעולה. כך למשל, הוראות המתייחסות לתכנית תשתמשנה באוגר מקטע CS, הוראות המתייחסות למחסנית תשתמשנה באוגר מקטע SS וכו').

המעבד יכול לאחסן מלה (word) של 16 סיביות בזיכרון בכתובת זוגית או אי זוגית לחילופין, ללא מגבלה כלשהי. במעבדים 8086, או 80286 ומעלה קיימת חשיבות לאחסון מלים בכתובות זוגיות. במעבדים אלה, אחסון המלים בכתובת זוגית מאפשר למעבד לקרוא נתון ברוחב 16 סיביות בגישה אחת לערוץ הנתונים. בגלל מבנה החומרה אחסון הנתון בכתובת אי-זוגית יחייב שתי גישות של המעבד לערוץ הנתונים ויאריך את זמן הביצוע של ההוראה.

נראה עתה כיצד משתמש המעבד באוגרי המקטע לצורך חישוב של כתובת אמיתית. כתובת אמיתית במרחב של 1MByte מורכבת משני אוגרים של 16 סיביות: אוגר מקטע (segment) ואוגר היסט (Offset). לצורך חישוב הכתובת האמיתית המעבד מחבר את התוכן של שני האוגרים בחיבור הזזה: המעבד לוקח את אוגר המקטע, מזיז אותו שמאלה בספרה הקסה-דצימלית אחת ומחבר אליו את אוגר ההיסט.

לדוגמה: נניח שאוגר DS מכיל כתובת 4500Hex ואוגר SI מכיל את הכתובת היחסית 0135Hex. כאשר המעבד ניגש לכתובת 135Hex יחסית למקטע הנתונים, מה תהיה הכתובת האמיתית?

צורת החישוב:	DS מוזז	45000 <-4500
		+
	תוכן SI נשאר	00135
	הכתובת במרחב הזיכרון	45135Hex

רישום של כל כתובת שהיא במרחב הזיכרון של המחשב ייעשה מטעמי נוחות והסכמה בצורה 4500:0135. ההצגה הינה תמיד הקסה-דצימלית, אלא אם נאמר אחרת. באותו אופן, כתובת 4513:0005 זהה לחלוטין לכתובת 4500:0135. בדוק !

1.8 אחסון מלה בזיכרון

הבית התחתון (LSB) של מלה יאוחסן בכתובת ראשונה ובית עליון (MSB) של המלה יאוחסן בכתובת הבאה. אחסון של כתובת בזיכרון בתוך משתנה המשמש כמצביע לכתובת תתבצע באופן דומה. כתובת ההיסט תאוחסן בכתובת ראשונה וכתובת אוגר המקטע תאוחסן בכתובת הבאה. בדוגמה לזה, אם מצביע בזיכרון מכיל את הכתובת 4500:0135, הוא יאוחסן בזיכרון כרצף של 4 בתים המתחילים בכתובת N, על פי הסדר הבא:

כתובת N	35
כתובת N + 1	01
כתובת N + 2	00
כתובת N + 3	45

תחילה המלה התחתונה LSW ולאחר מכן המלה העליונה MSW. בתוך כל מלה אופן האחסון יהיה כפי שהוסבר לגבי אופן אחסון מלה בזיכרון.

בתוך מרחב הזיכרון ישנם איזורים מסוימים הנשמרים לצורך ביצוע פעולות ייחודיות למעבד:
כתובות 00000-003FFHex (1024 כתובות) שמורות לצורך פעולות פסיקה. מנגנון הפסיקה יוסבר בהמשך, אך נאמר כעת, כי עבור וקטור המכיל כתובת פסיקה דרושים 4 בתים: כתובת אוגר מקטע (2 בתים) וכתובת אוגר היסט (2 בתים). מכאן שמרחב הפסיקות מכיל מקום ל-256 וקטורי פסיקה שונים.

כתובות FFFF0-FFFFFHex (16 כתובות) שמורות לקוד המעביר את הפיקוח על פעולות המעבד לקטע תכנית המכיל אתחול למערכת. פעולת Reset המתרחשת עם הדלקת המחשב או לחיצה על Reset גורמות למעבד להתחיל את ביצוע ההוראות מכתובת FFFF:0000Hex.

1.9 פעולת Reset במעבד

אתחול של המעבד או הדלקה של המערכת נעשים על ידי הרמה של הדק Reset ("רגל" של הרכיב) במעבד ל-"1" והורדתו חזרה ל-"0". פעולה זו תגרום למעבד להכניס את הערך FFFFHex לאוגר המקטע של התכנית (CS) ואת הערך 0000Hex לאוגר ההוראות (IP). לכן, המעבד יתחיל לבצע הוראות הכתובות בקטע התכנית המתחיל בכתובת FFFF:0000Hex (כתובת FFFF0). פעולת אתחול זו מוחקת את כל המידע הקודם הקיים באוגרים שבמעבד ומאפשרת להתחיל תהליך חדש במערכת ללא תלות במצב קודם. כאשר מדליקים את המחשב האישי מתבצעת פעולת BOOT, במסגרת פעולה זו מתבצעת גם פעולת RESET למעבד.

1.10 פסיקות (Interrupts)

כאשר המעבד עסוק בביצוע תכנית מסוימת אין באפשרותו לבדוק כל הזמן אם התקן מסוים מבין כל ההתקנים המחוברים אליו מבקש ממנו שירות כלשהו. אם כל תכנית היתה מנסה לבדוק כל הזמן את כל האפשרויות הקיימות לבקשות שירות מהתקנים, חלק ניכר מזמן המחשב לא היה מנוצל לצורכי עבודה אלא לצורכי בדיקות אלו. כדי למנוע זאת, משתמש המעבד במנגנון הפסיקה. באמצעותו המעבד אינו צריך לבדוק כל הזמן בקשות לשירות והוא עסוק כל הזמן בביצוע התכנית.

כאשר התקן מסוים זקוק לשירות של המעבד במהלך הפעילות, הוא **"מפריע"** למעבד על ידי סימון מוסכם. בשלב זה המעבד מפסיק (ומכאן המונח "פסיקה") את הפעילות שבה עסק בזמן שזיהה את בקשת הפסיקה ופונה לטפל בהתקן שביקש פסיקה זו. המעבד אינו מפסיק את פעולתו סתם כך, אלא שומר את מצב האוגרים ושאר הערכים הדרושים על מנת לאפשר חזרה בצורה תקינה ואמינה לפעילות שהופסקה.

כאשר תכנית של המשתמש מבקשת שירותים של מערכת ההפעלה היא משתמשת בפסיקות שהוגדרו במפרט הטכני של מערכת ההפעלה, מבלי לדעת היכן נמצאת בפועל תכנית הפסיקה. זהו יתרון גדול מבחינת גמישות השימוש של המערכת

במנגנוני פסיקה, אשר גם ניתן להגדירם באופן דינמי על ידי אתחול וקטורי הפסיקות בזיכרון RAM הקשור למעבד.

פסיקה הינה תהליך שבאמצעותו "מפסיקים" את הפעולה השוטפת של המעבד לצורך ביצוע פעולה אחרת, תוך כדי שמירת הכתובת הקודמת שבה היה המעבד בעת ביצוע הפסיקה. האוגרים הנשמרים הינם CS ו-IP המצביעים על ההוראה הבאה שעל המעבד לבצע כאשר יסיים את הטיפול בבקשת הפסיקה.

את הפסיקות ניתן לחלק לשני סוגים: **פסיקות חומרה ופסיקות תוכנה** (Software & Hardware Interrupts).
ההבדל בין שני הסוגים נובע מהשאלה מי יוזם את בקשת הפסיקה מהמעבד? בפסיקת חומרה יוזם את הפסיקה גורם חיצוני למעבד, או תהליך במעבד שהתכנית אינה שולטת עליו, או שאינה מושפעת ממנו (חלוקה באפס, הרצה של צעד אחרי צעד – Single Step). את פסיקות התוכנה יוזמת התכנית, על מנת לקבל שירות ממערכת ההפעלה, או מתכניות שירות הקיימות במערכת המחשב. פסיקת תוכנה מאפשרת למשתמש לקרוא לתכנית שירות אשר אין הוא יודע את כתובתה.

פסיקות חומרה נחלקות לשני סוגים: **פסיקות שניתנות למיסוך ופסיקות שאינן ניתנות למיסוך** (Maskable and NON maskable). בפסיקות שאינן ניתנות למיסוך המתכנת אינו יכול לחסום את ביצוע הפסיקה והיא תתבצע בכל מקרה (דוגמה לפסיקה כזו היא הדק NMI במעבד). בפסיקות הניתנות למיסוך, המתכנת יכול לחסום את הביצוע שלהן ולאפשר אותן שוב כשהדבר נחוץ. על הדק Intr במעבד ניתן לשלוט בתוכנה באמצעות הפקודה Sti והפקודה Cli.

תהליך הביצוע של פסיקה

במשפחת המעבדים 86 ניתן להגדיר עד 256 פסיקות שמספרן 00-FFHex. כאשר נגרמת פסיקה על ידי גורם חיצוני או פנימי ומנגנון הפסיקות של המעבד מאפשר למעבד לקבל בקשת פסיקה, הוא פועל כך:
מסיים את ביצוע ההוראה הנוכחית ושומר במקטע המחסנית את אוגר הדגלים ואת כתובת ההוראה הבאה שיש לבצע בגמר תכנית הפסיקה. רק האוגרים CS, IP ו-FLAGS נשמרים באופן אוטומטי. תכנית הפסיקה אחראית לשמירת אוגרים נוספים שבהם היא משתמשת.

המעבד פונה לווקטור הפסיקות, מחשב את כתובת תכנית הפסיקה ומעדכן את תוכן האוגרים CS, IP ועובר לביצוע של תכנית הפסיקה. כתובות תכניות הפסיקה נמצאות במערך פסיקות המכיל 00-FFHex (עד 256) וקטורים בכתובות 0000-03FFHex של זיכרון RAM, אשר משמשות לצורך זה בלבד. כתובת תכנית פסיקה מורכבת משתי מלים בנות 16 סיביות שבהן כתובת היסט וכתובת מקטע. כאשר המעבד שולף כתובת זו, הוא מכניס את כתובת ההיסט לאוגר IP ואת כתובת המקטע לאוגר CS וממשיך את פעילות המערכת מכתובת תכנית הפסיקה.

על מנת לאפשר למעבד לחזור לביצוע התכנית אשר הפסיק כאשר פנה לביצוע תכנית הפסיקה, תכנית הפסיקה צריכה להסתיים בפקודה Iret. פקודה זו מורה למעבד לעדכן את האוגרים CS, IP, FLAGS על פי הערכים שנשמרו במקטע המחסנית לפני שהוא פנה לביצוע תכנית הפסיקה. עדכון האוגרים מאפשר למעבד להמשיך בפעילות שהפסיק בלא שיורגש שינוי בביצוע התקין של התכנית שהופסקה (פרט למשך זמן הביצוע כמובן).

1.11 גישה למרחב הקלט/פלט

מרחב הקלט/פלט (Input/Output - I/O) מהווה תוספת למרחב הזיכרון במעבד. המעבד ניגש למרחב זה על ידי הוצאת כתובת ברוחב 16 סיביות אל ערוץ הכתובות (הדקים A0-A15 של ערוץ הכתובות המגדירים 65536 צירופי כתובות שונים). המעבד מאפשר למתכנת לשלוט על מרחב זיכרון זה באמצעות שתי פקודות מיוחדות השייכות לאוסף הפקודות של המעבד: הפקודות IN ו-OUT מאפשרות להעביר נתונים מהמעבד למרחב הקלט/פלט ולהיפך. בפעולות אלו משתתף אוגר AL להעברת נתונים ברוחב 8 סיביות, או אוגר AX להעברת נתונים ברוחב 16 סיביות. אל כתובות 0 עד 00-FFHex במרחב הקלט/פלט ניתן לגשת במיעון ישיר. גישה לכתובות 100Hex ואילך במרחב הקלט/פלט נעשית באמצעות אוגר DX שערכו מציין כתובת במרחב הקלט/פלט.

1.12 תכונות מעבד 80286

מעבד 80286 הופך למעבד בסיסי במחשבים האישיים (מעבד 8088 מפנה מקומו ל-80286, תוך כדי דילוג על דורות הביניים 8086 ו-80186). למעבד 80286 מספר תכונות נוספות על מעבדי 8088/8086 המקנות לו ביצועים טובים יותר ואפשרויות עבודה רחבות יותר. למעבד 80286 ערוץ כתובות ברוחב של 24 סיביות וערוץ נתונים של 16 סיביות המאפשר גישה למרחב כתובות גדול יותר והעברת נתונים רבה יותר מאשר במעבד 8088. למעבד 80286 שני אופני פעולה שונים:

1.12.1 מצב אמיתי - Real Mode

באופן עבודה זה המעבד מתנהג בדיוק כמו מעבד 8088, אך עם ערוץ נתונים של 16 סיביות. באופן עבודה זה הגישה של המעבד מוגבלת למרחב כתובות של 1MByte בלבד, כמו במעבד 8088 ואין אפשרות לגשת ולנצל את שאר המרחב האפשרי במחשב. כל תכנית המתבצעת במעבד יכולה לגשת לכל כתובת במרחב הכתובות ולבצע כל פקודה מאוסף הפקודות של המעבד ללא הגבלה כלשהי, בדיוק באותו אופן שבו היא מתבצעת במעבד 8088. מעבד 80286 מבצע את התכניות במהירות גבוהה יותר ממעבד 8088 מכמה סיבות:

* מעבד 80286 יכול לעבוד בתדרים גבוהים יותר ממעבד 8088.

* המבנה הפנימי של מעבד 80286 משוכלל יותר ומכיל יותר שלבים במסלול הצינור Pipe-Line (ארבעה שלבים לעומת שנים בלבד במעבד 8088).

* המעבד יכול להעביר בפקודת מכונה אחת נתונים ברוחב 16 **סיביות** ולכן הזמן הנדרש להעברת כמות נתונים שווה קצר יותר במעבד 80286.

1.12.2 מצב מוגן - Protected Mode

באופן עבודה זה ניתן לנצל את תכונות המעבד אשר לא באות לידי ביטוי בצורת עבודה Real Mode. באופן עבודה זה יכול המעבד לגשת ל-16MByte של מרחב הכתובות. בנוסף לכך, למעבד 80286 ישנה היכולת לבצע מספר תכניות במקביל ועל ידי כך לנצל בצורה יעילה יותר את המעבד. אם היינו מנסים לבצע מספר תכניות במקביל באמצעות מעבד 8088, התכניות היו עלולות להפריע זו לזו. למעבד 8088 אין תכונות אשר מאפשרות למערכת ההפעלה להגן על מרחב הכתובות של תכנית אחת בפני תכניות אחרות. כל תכנית שולטת למעשה על כל מרחב הזיכרון ויכולה לגשת לכל כתובת במרחב זה. לעומת זאת, במעבד 80286 ישנן מספר תכונות המאפשרות לו לבצע מספר תכניות במקביל, להגן על מרחב זיכרון של תכנית בפני גישות של תכנית אחרת ועם זאת - לשתף קטעי זיכרון בין תכניות. התכונות המיוחדות נחלקות למספר סוגים:

א. הגנה - Protection

תכונה זו מאפשרת לתחום כל תכנית בקטע זיכרון המגדיר תחום מותר ואסור לגישה על ידי תכנית זו. תיחום זה של התכנית מאפשר לה לפנות אל האיזורים המותרים לה, ואינו מאפשר לה לגשת לאיזורים בזיכרון השייכים לתכניות אחרות המתבצעות המערכת. הוא מאפשר הגנה ובידוד של תכניות. לכל תכנית מוגדר תחום כתובות על ידי **אוגרי תחום**. כל גישה של התכנית לכתובת במרחב הכתובות, נבדקת על פי ערכם של אוגרי התחום המוגדרים לה ואם הכתובות המבוקשת נמצאת בתחום, המעבד יאפשר לגשת לתחום זה והוא יאסור גישה לכתובת אם איננה בתחום המותר.

תכונת ההגנה מאפשרת ליצור על מערכת המבוססת סביב מעבד 80286, מערכת של ריבוב משימות (Multi-Tasking). כלומר, ריצה של מספר תכניות במקביל במערכת מחשב אחת. דוגמה לכך יכולה להיות הרצה של תכנית שמבצעת חישובי משכורת ובמקביל לעבוד על מעבד תמלילים בלא שתכנית חישוב המשכורות תפריע לעבודה על מעבד התמלילים ולהיפך. ביצוע תכניות במקביל על גבי מעבד יחיד במערכת אינו יכול להיות בו זמנית. עלינו לזכור שבמערכת קיים מעבד יחיד, המבצע את הפקודות בזו אחר זו. עם זאת, הוא מאפשר להפסיק פעולה של תכנית, לעבור לתכנית אחרת לפרק זמן מסוים ולחזור לתכנית הקודמת. המעבר חייב להיעשות בצורה תקינה על מנת שהמשתמש לא ירגיש בו.

המעבד 80286 מספק כלים למעבר מביצוע תכנית אחת לחברתה על ידי שמירת מצב מלאה של התכנית המופסקת והצבה של מנגנוני הגנה המאפשרים לה להתבצע ללא הפרעות של תכניות אחרות. המעבר מתכנית לתכנית נעשה בעזרת תוכנה השולטת על המערכת, כאשר פרק הזמן הנדרש על מנת לבצע מעברים אלה הולך ועולה ככל שיש לבצע יותר תכניות. מספר התכניות שניתן לבצע תלוי בתוכנה המפעילה את ריבוב המשימות.

ב. זיכרון בפועל - Virtual Memory

תכונת זיכרון בפועל מאפשרת למעבד להקצות לכל תכנית עד 1GByte של זיכרון, כאשר הזיכרון בפועל שאליו המעבד 80286 יכול לגשת הינו 16MByte. תכונה זו מאפשרת להשתמש בדיסק המגנטי המחובר למערכת המחשב גם כשטח עבודה של התכנית בנוסף להיותו הזיכרון הראשי של

המעבד. זהו היישום של **זיכרון בפועל** (Virtual Storage) – היכולת להציג זיכרון גדול יותר מזה שקיים למעשה במחשב. במקרה כזה הרווח בגודל זיכרון בא על חשבון מהירות הגישה לנתונים, מכיון שחלק מהם נשמרים על גבי דיסק מגנטי שמהירות הגישה אליו נמוכה בהרבה ממהירות הגישה לזיכרון RAM של המחשב. שינוי זה במהירות הגישה גורם להאטה מסוימת בביצוע התכניות.

הזיכרון בפועל מושג על ידי חלוקת הזיכרון בצורה שונה מהחלוקה הרגילה של המעבד. חלוקה רגילה מאפשרת למעבד גישה לנתונים ברוחב 16 או 8 סיביות בלבד. חלוקה זאת נקראת חלוקה למקטעים (Segments).

ג. מקטעים - Segments

המקטעים הינם קטעי זיכרון אשר מתקבלים מחלוקה של הזיכרון למספר חלקים בעלי גודל משתנה בהתאם לצרכים. את המקטעים הללו ניתן לשמור בזיכרון RAM של מערכת המחשב, או בדיסק מגנטי והמעבד יכול להבחין ולזהות היכן שמור המקטע. המעבד מאפשר תוך כדי ביצוע של התכניות לטעון, או לשמור, מקטעים בצורה אוטומטית בזיכרון הראשי, או בדיסק המגנטי.

תכונה זאת של המעבד 80286 מאפשרת ליישם זיכרון בפועל בשיטה החלוקה למקטעים (Segmentation). המעבד יכול לשתף מקטעים בין תכניות ועל ידי כך לחסוך במקום אחסנה בזיכרון הראשי. שיתוף של קוד בין תכניות (Shared Memory) מאפשר למספר משתמשים הפועלים על אותה תכנית, להחזיק רק עותק יחיד של תכנית זו בזיכרון הראשי ובכך לחסוך מקום רב בזיכרון זה.

למרות כל היתרונות של מעבד 80286, **במצב הקיים כיום תחת מערכת ההפעלה DOS, אין אפשרות לניצול מלא של תכונותיו.** יש למפתחים מחויבות לתאימות עבודה כלפי מטה של מערכת ההפעלה DOS ושל היישומים שנבנו סביבה עבור המעבדים 8086 ו-8088. לפיכך יש קושי בהתאמתה לניצול מרחב הזיכרון הגדול יותר של מעבד 80286 ושאר התכונות המאפשרות ריבוב משימות.

תכניות הנכתבות עבור מעבד 8088 אינן מתחשבות במגבלת ההגנה בין תכניות. הן פונות ישירות לכל מרחב הזיכרון, או לכרטיסים השונים הקיימים במערכת, ולכן לא ניתן להמיר אותן בצורה אוטומטית לעבודה באופן העבודה המוגן של מעבד 80286. עם זאת, קיימת אפשרות לניצול של הזיכרון הגדול יותר של מעבד 80286 (Extended Memory).

מערכות הפעלה המנצלות את כל תכונות המעבד הן מערכות הפעלה שפותחו עם גישה לריבוב משימות כדוגמת מערכת ההפעלה OS/2 של יבמ, מערכות הפועלות תחת מערכת הפעלה UNIX וכו'. מערכת מחשב אישי המבוססת על מעבד 80286 ופועלת באותו התדר שבו פועלת מערכת מחשב המבוססת על מעבד 8088, תבצע את אותה התכנית בחצי הזמן לערך. למעבד 80286 ניתן לחבר מעבד מתמטי 80287 המאפשר ביצוע פעולות על נתונים ממשיים באמצעות חומרה.

1.13 המעבדים 80386SX, 80386DX

המעבדים ממשפחת 80386 בנויים בארכיטקטורה של **32 סיביות**. כלומר, רוחבו של כל אוגר במעבד הוא 32 סיביות וניתן לבצע בחומרה פעולות על נתונים ברוחב זה. הארכיטקטורה החיצונית של שני המעבדים שונה כאשר מסתכלים על הערוצים החיצוניים שלהם.

מעבד 80386SX מתקשר לסביבה החיצונית באמצעות ערוץ כתובת ברוחב של 24 סיביות וערוץ נתונים של 16 סיביות, **בדיוק כמו מעבד 80286**. צורת עבודה זו מאפשרת תאימות חומרה כלפי מטה בכרטיסים שנועדו למחשבים אישיים עם מעבדים קודמים. הארכיטקטורה הפנימית של 32 סיביות מאפשרת ל-80386SX להנות משני העולמות: להריץ תכניות שנכתבו עבור מעבדי 8088/8086 ועבור מעבדי 80286, ובכל זאת להנות מכל התכניות המיועדות לארכיטקטורה של 32 סיביות.

מעבד 80386DX פועל בארכיטקטורה פנימית וחיצונית ברוחב של 32 סיביות. כלומר, הוא מתקשר לסביבה החיצונית לו על ידי ערוצי כתובת ונתונים ברוחב של 32 סיביות. למעבד 80386DX כל התכונות של מעבד 80286 ונוספות להן אחרות. גם על מעבדים אלה ניתן לבצע תכניות שמיועדות עבור מעבדי 8088/8086 ו-80286.

מעבד 80386 יכול לפעול **באופן פעולה מוגן** (Protected Mode) **ובאופן פעולה אמיתי** (Real Mode) כמו מעבד 80286. למעבד 80386 ניתן לחבר מעבד עזר מתמטי 80287 או מעבד עזר מתמטי 80387 (מעבד עזר 80387SX עבור מעבד 80386SX), שהינו מהיר יותר ומאפשר האצה של מהירות ביצוע התכניות. אפשר לחבר למעבד 80386 זיכרון מטמון (Cache) מהיר ועל ידי כך להגביר עוד יותר את ביצועיו (הסבר על זיכרון מטמון ראה בהמשך).

מעבדי 80386 מאפשרים לגשת לנתונים בזיכרון ברוחב של 8, 16 ו-32 סיביות. בנוסף לטיפוסי נתונים אלה ניתן להגדיר טיפוסי נתונים ברוחב גדול יותר כמו **דפים** (Pages) **ומקטעים** (Segments, כמו במעבד 80286). הדפים הם קטעי זיכרון בני 4KByte כל אחד. כל דף ניתן לשמור בזיכרון ראשי או על גבי דיסק מגנטי ולשתף אותו בין תכניות יישומיות שונות. בדרך זו יכולה מערכת ההפעלה לבנות מערכת **דפדוף** (Paging). פעולת הדפדוף חיונית למערכת המאפשרת הפעלה של מספר תכניות במקביל בשיטת זיכרון בפועל (Virtual Memory).

תכונות אלו של המעבד הכרחיות למערכת ההפעלה התומכת באפשרות ביצוע של מספר תכניות במקביל. הן מאפשרות למערכת ההפעלה ל"ראות" זיכרון ראשי גדול מאוד לעומת הזיכרון הראשי הפיסי. היא משתמשת בחלק מהדיסק הראשי לצורך שמירת חלקים של תכניות המתבצעות בזיכרון ובכך היא יכולה להוריד את מגבלת גודל הזיכרון הראשי. עם זאת ראוי לזכור ש"נשלם" בזמן ביצוע איטי יותר בהבאה של מקטעים או דפים מהדיסק.

בנוסף לאופני העבודה המוגדרים במעבד 80286, יכולים מעבדי 80386 לפעול בשיטת **זיכרון בפועל** (Virtual Mode), המאפשרת למספר תכניות שנכתבו למעבד 8088 לרוץ במקביל ובלא הפרעה זו לזו, באופן שכל תכנית מרגישה שהיא מתבצעת על גבי מעבד 8088/8086 נפרד. כמו במעבד 80286 ניתן לעבור

גם כאן מביצוע של תכנית אחת לחברתה באמצעות פקודה. אך כאן, המעבר מתכנית לתכנית נעשה בחומרה ועל כן הוא מהיר יותר. עבור כל תכנית שומר המעבד אוגרים נפרדים, ועל כן מהירות המעבר מתכנית לתכנית גבוהה יותר ואמינה יותר. מכיון שהמעבר בין תכניות נעשה בחומרה והמעבד מקצה קבוצת אוגרים עבור כל תכנית, קיימת מגבלה ביצוע של 16 תכניות במקביל.

מהירות הביצוע של תכניות במקביל במעבדי 80386 גבוהה בהרבה ממהירות הביצוע על מחשבים מבוססי 80286. תכונה זאת המאפשרת לכל משימה לראות מחשבי מושלם משלה ומאפשרת הפרדה מלאה בין תכניות. חלוקה כזאת קיימת בעיקר במערכות מחשבים גדולות של יבמ ונקראת **מחשב בפועל** (Virtual Machine).

מעבדי 80386 מאפשרים חיבור לזיכרון מטמון (Cache). זיכרון המטמון הינו בדרך כלל זיכרון RAM סטטי קטן ומהיר ביותר (מהירותו 10-40nSec לעומת מהירות זיכרון RAM דינמי שהינו 70-100nSec) המחובר למעבד באמצעות בקר מתאים. זיכרון המטמון קטן יחסית לזיכרון הראשי ומהווה חיץ בינו לבין המעבד. זיכרון RAM הראשי במערכת אינו מסוגל לעבוד בקצב העבודה של המעבד ותמיד הוא יהיה איטי יותר ממנו. הסיבה הינה כלכלית - המחיר של הזיכרון הראשי, לו רצו להעלות את מהירותו.

זמן התגובה האיטי יחסית של זיכרון RAM הדינמי הראשי לעומת מהירות המעבדים החדשים והמהירים אינו מאפשר לו לפעול במהירות שבה הם פועלים. זיכרון המטמון המהיר יותר יכול להאיץ את פעולת המערכת בהרבה. גם זיכרון המטמון מנצל את תופעת המקומיות. המעבד שומר בזיכרון המטמון נתונים וחלקי תכניות שבהם ישתמש בזמן הקרוב, כך שבאחוז גבוה של המקרים הוא ניגש לזיכרון המטמון המהיר במקום לגשת לזיכרון RAM הדינמי האיטי יותר. רוב פעילות המחשב מתבצעת מול זיכרון המטמון המהיר, כך שמהירות הביצוע של תכנית המערכת המבוססת על מעבדי 80386 וזיכרון מטמון גבוהה בהרבה מביצועי מערכת ללא זיכרון מטמון.

תכונות אלו הופכות את מעבדי 80386 לנדרשים יותר ממעבדי 80286. מספר גדל והולך של תכניות נכתבות כיום במיוחד למעבדי 80386 ועל כן לא תוכלנה לפעול על מעבדים קודמים. מערכות המיועדות לפעול עם מספר יישומים במקביל יש לבסס על מעבדי 80386 כתקן מינימלי. ביצועי מערכת מחשב מבוססת 80386 גבוהים יותר מזו של מערכת מבוססת 80286 ואמינות פעולתה גבוהה יותר.

1.14 מעבד 80486

מעבד זה הינו המתקדם ביותר הקיים היום בסדרת המעבדים של Intel בארכיטקטורה זו. מעבד 80486 הינו בעל ביצועים גבוהים ביותר, העולים בכמה סדרי גודל על הנדרש ממחשב ביתי. המעבד בנוי בארכיטקטורה של 32 סיביות מלאה (ערוצי כתובות ונתונים ברוחב 32 סיביות) ומסוגל לפעול במהירות גבוהה יותר ממעבדי 80386.

בנוסף לתכונות הקיימות במעבדי 80386, מספק מעבד 80486 בתוך אריזה אחת מערכת מחשב הכוללת בקר זיכרון מטמון וזיכרון מטמון בגודל 8KByte,

מעבד מתמטי 80387 שהינו חלק אינטגרלי של המעבד (ולא רכיב נפרד). רכיבים אלה ששולבו במעבד מאפשרים לו לפעול בכ-50 אחוזים יותר מהר ממעבד 80386 מקביל. על כן, התפוקה של מערכות מבוססות 80486 הפועלות במהירות 25MHz תפעלנה הינה גבוהה יותר מזו של מערכות מבוססות 80386 הפועלות במהירות 33MHz. בדרך כלל התפוקה תהיה כ-10-20% יותר, בהתאם לסוג התכניות. במעבד מסוג 80486 יש כ-1.2 מיליון טרנזיסטורים הארוזים ביחידה אחת, לעומת כ-275 אלף ב-80386.

מעבד 80486 מאפשר ביצועים של מערכת מיני מחשב. מבנה המעבד מאפשר להקים מערכת מחשב מהירה ביותר, קטנה ופשוטה המכילה באריזה אחת את מירב התקני החומרה שהמשתמש זקוק להם. תכונה זאת באה לפתור מצב קודם שבו בנייה של מערכת מחשב בתצורה בסיסית חייבה חיבורים ותוספות של התקנים חיצוניים למעבד שסופקו כאילתורים שונים על ידי יצרני חומרה שונים. תצורה תקנית זאת מאפשרת להפוך את מערכות המחשב לתואמות יותר זו לזו כדי לתת פתרונות תקניים המקלים על יצרנים ומשתמשים כאחד.

מעבד 80486 מתאים בעיקר לצרכים של מפתחי חומרה למיניהם מכיון שעיקר הגישה הינה למזעור הרכיבים והכנסת יותר רכיבים חיצוניים פנימה. אריזה זאת מאפשרת לבנות מערכות פשוטות יותר עם פחות התקנים חיצוניים. גירסאות ראשונות של מעבד זה נתקלו בבעיות קשות והיו בהם תקלות תוכנה שונות, אשר רובן נבעו מהסיבוכיות הרבה ביישום אוצר הפקודות של המעבד על ידי בניית **יחידת בקרה** (Control Unit) של בשיטת Discrete Logic (מעגלים אלקטרונים ייעודיים) ולא בשיטת Micro Program המקובלת יותר במעבדים מסדר גודל זה.

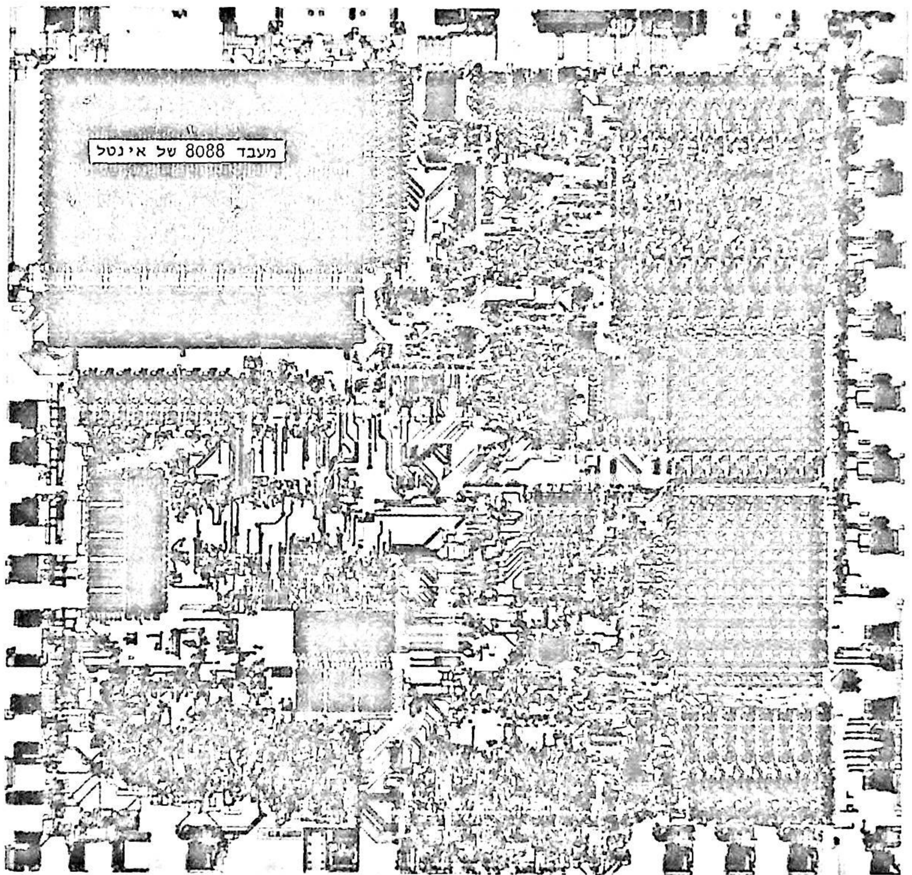
מעבד 80486 אינו מוסיף פקודות רבות לאוסף הפקודות של מעבדי 80386, והתוספת של המעבד המתמטי כחלק של המערכת תורמת רק למשתמשים העוסקים הרבה בחישובים מתמטיים ובחישובי פיסיקה, שרטוט ותיב"ם. משתמשים רגילים ביישומים מינהליים יוכלו להפיק תועלת מעטה, כי לא כל תכנית צריכה או יודעת להשתמש במעבד המתמטי). מכאן, שתוספת המעבד המתמטי בתוך החבילה הכוללת אינה משפרת את ביצועי המערכת בהרבה עבור משתמשים אלה. למרות שהמעבד המתמטי קיים כחלק של המעבד, משתמשים הזקוקים למעבדים מתמטיים חזקים ומהירים יותר, יכולים לחבר מעבד מתמטי נוסף, כמו למשל Weitek 4167.

מהירות הביצוע הגבוהה יותר של מעבד 80486 ביחס למעבדי 80386 תורגש בשיטת העבודה המוגנת שעבורה נבנה המעבד, ולא בשיטה שבה פועלת מערכת ההפעלה DOS (שיטת Virtual).

כל משפחת מעבדי 86 של חברת Intel מספקת תאימות תוכנה כלפי מטה. כלומר תכניות שנבנו במקורן למעבד מסוים ירוצו ויתבצעו גם על מעבדים מתקדמים יותר, אשר ישמרו על תאימות תוכנה כלפי מטה. תכניות המנצלות את התכונות הייחודיות של המעבדים המתקדמים לא יוכלו לפעול במעבדים מדור קודם. הפעלה יעילה של המעבדים מחייבת התאמה של התכניות לאופני הפעולה המתקדמים יותר המוצעים על ידי יצרן המעבד.

1.15 טבלאות להשוואת תכונות מעבדים

התכונה	8088	8086
רוחב ערוץ נתונים	8	16
רוחב ערוץ כתובות	20	20
מרחב זיכרון פיסי	1MByte	1MByte
מרחב I/O	64KByte	64KByte
מעבד מתמטי	8087	8087
תדרי עבודה MHz	5-12	5-12


מעבד 8088 של אינטל

התכונה	80286	80386SX	80386DX	80486
רוחב ערוץ נתונים	16	16	32	32
רוחב ערוץ כתובות	24	24	32	32
מרחב זיכרון פיסי	16MByte	16MByte (באופני עבודה מתקדמים בלבד)	4GByte	4GByte
מרחב זיכרון Real Mode תואם 8086/8088	1MByte	1MByte	1MByte	1MByte
מרחב זיכרון Virtual Real Mode תואם 8086/8088	לא קיים	1MByte per task	1MByte per task	1MByte per task
מרחב I/O	64KByte	64KByte	64KByte	64KByte
מעבד מתמטי	80287	80387SX 80287	80387 80287	קיים מעבד 80387 בתוך המעבד
תמיכה במקטעים בנוסף לבית ומלה. אפשרות ניהול זיכרון בשיטת Segmentation	יש	יש תכונה זו מאפשרת לבנות מערכת הפעלה המיישמת Virtual Memory	יש	יש
תמיכה בדפים אפשרות ניהול זיכרון בשיטת Paging	אין	יש תכונה זו מאפשרת ליישם Virtual Memory בשיטת Paging	יש	יש
זיכרון מטמון (Cache)	אין	חיצוני למעבד (כולל בקר חיצוני)	חיצוני למעבד	8KB בתוך המעבד כולל בקר
תדרי עבודה, MHz	6-20	16-20	16-40	25-50

פרק 2

הלוח הראשי במחשב האישי

הלוח הראשי (Mother Board) במחשב מהווה את לב המערכת. הוא מכיל מספר התקנים ורכיבים, אשר אינם מספיקים להפעלת המחשב האישי. כדי שהלוח יהפוך למערכת מחשב יש להוסיף לו התקנים שונים באמצעות מחברי ההרחבה שנמצאים עליו. במחברים אלה נועצים את כרטיסי תיאום אשר מהווים את התוספות לתצורת החומרה הבסיסית. וכך, הלוח הראשי וההתקנים הקשורים אליו מהווים מערכת מחשב מינימלית.

בלוחות ראשיים של מערכות PS/2 ו-PS/1 של IBM ובמספר לוחות ראשיים המבוססים על מעבדי 80386, ניתן למצוא על גבי הלוח הראשי את כל כרטיסי החומרה הנדרשים להפעלת המערכת. בלוחות ראשיים אלה אין צורך להוסיף מתאמי חומרה נוספים על אלה הקיימים. עם זאת, יש לזכור כי תקלה בתפקוד של אחד המתאמים הנמצא על גבי הלוח הראשי תחייב בדרך כלל את החלפת הלוח הראשי כולו, גם אם האמינות הכוללת של הלוח גבוהה יותר.

במחשבים תעשייתיים נמצא את הלוח התחתון (שהוא בדרך כלל הלוח הראשי) ריק כמעט לחלוטין. הלוח התחתון, שנקרא Back Plan, מכיל אך ורק ספק מתח ואת קווי הערוצים השונים במערכת (ערוץ - Bus). במחשבים אלה המעבד ושאר חלקי המחשב הפנימיים שנמצאים בדרך כלל על הלוח הראשי מותקנים על כמו כל כרטיס מתאם אחר באחד ממחברי ההרחבה הקיימים בלוח הראשי.

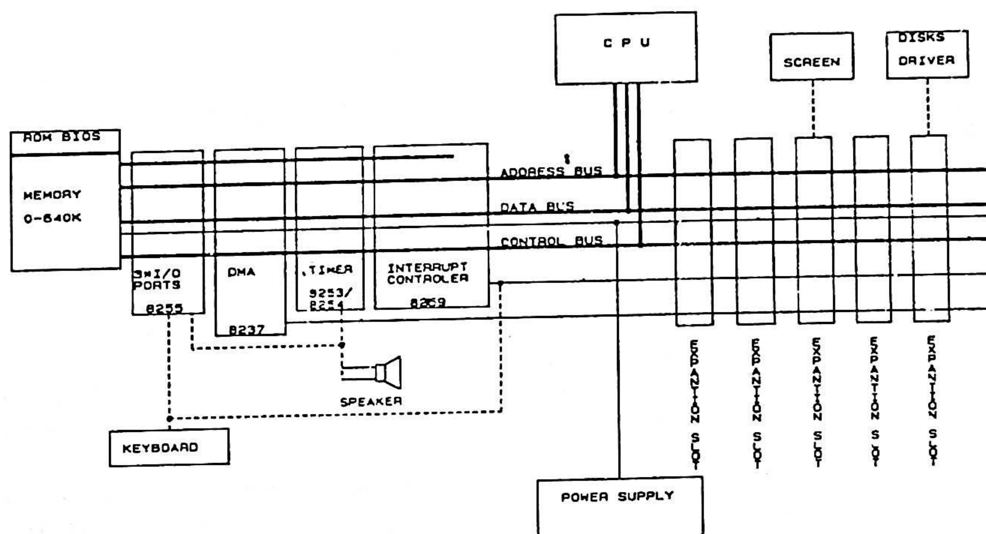
2.1 מבנה כללי של הלוח הראשי במחשב

סכימה של הלוח הראשי מוצגת בתרשים המלבנים המצורף. נסקור את מרכיביו.

המעבד מהווה את מרכז הלוח הראשי. הוא מנהל שלושה ערוצים המחוברים גם לרכיבים בלוח הראשי וגם למחברי ההרחבה המאפשרים חיבור של כרטיסי תיאום חיצוניים. על גבי הלוח הראשי יש מקום להכנסת זיכרון RAM בקיבולת עד 640KByte במחשב מדגם XT. במחשבים המבוססים סביב מעבדים מתקדמים אפשר להתקין זיכרון גדול יותר. על הלוח הראשי נמצא גם זיכרון ROM המכיל את תוכנת BIOS. גודלו של זיכרון זה הינו לפחות 8KByte.

ארבעה רכיבים מרכזיים נוספים נמצאים על הלוח הראשי: 8255, 8237, 8259 ו-8253. על תפקידיהם ותכונותיהם של רכיבים אלה נדון בהמשך. מארז המחשב מכיל גם רמקול (להשמעת צפצוף), אשר מתחבר לרכיב 8253, ותכנית המתבצעת במחשב יכולה לשלוט על פעולתו.

המקלדת, או לוח המקשים, נמצאת במארז נפרד ומתחברת אל הלוח הראשי באמצעות מחבר הנמצא באחורי הלוח הראשי. בכמה דגמים החיבור הוא מלפנים, אך אין לכך כל חשיבות. לחיבור המקלדת אין צורך בכרטיס תיאום, כי כל החומרה הנדרשת להפעלתה קיימת על גבי הלוח הראשי. המסך במחשבי XT ו-AT מתחבר בעזרת כרטיס תיאום אשר מורכב באחד ממחברי ההרחבה. ספק הכוח נמצא במארז המחשב ומספק את מתח ישר ברמות +5V, -5V, +12V, -12V. הספק נדרש להפעלת הלוח הראשי, כונני הדיסקים/דיסקטים ושאר כרטיסי התיאום המחוברים למערכת המחשב. כרטיסי התיאום השונים המחוברים למערכת המחשב מקבלים את המתח הנדרש להפעלתם דרך מחברי ההרחבה.



הלוח הראשי - תרשים מלבני

2.2 מחשבים תואמים

כאשר יבמ נכנסה בשנת 1981 לשוק המחשבים האישיים היא פעלה בשיטה של ארכיטקטורה פתוחה. משמעות הארכיטקטורה הפתוחה היא במסירת כל המידע הנדרש ליצרני חומרה ותוכנה כדי שיוכלו לפתח התקני חומרה למערכת ותכניות שירות שונות. יבמ סיפקה נתונים מפורטים ומדויקים על מבנה החומרה והתוכנה ובכך אפשרה תפוצה רחבה ביותר של המחשב האישי ותרמה למהפך הגדול בהיקף התפוצה שלו. התוצאה השלילית מבחינתה של יבמ היתה התחרות. שוק המחשבים האישיים גדל, המכירות גדלו ויצרני חומרה רבים נכנסו לשוק זה ויצרו מחשבים שהיוו חיקוי מושלם למערכת המחשב האישית של יבמ.

מחשבים תואמים אלה מאופיינים בכך שהם מסוגלים לבצע כל מה שמערכת מחשב יבמ יודעת לבצע. כלומר, תכנית תתבצע באותו אופן על מחשב תואם או במחשב יבמ. כל כרטיס תיאום חומרה ניתן לחבר באופן שיפעל בצורה תקינה על מחשב תואם או במערכת יבמ מקורית. ארבעה הבדלים קיימים בין מערכות תואמות שונות והם מדורגים על פי סדר חשיבות:

* מהירות הביצוע של המערכת.
* העיצוב החיצוני והמכני של מערכת המחשב.
* טיב הייצור וההרכבה של הלוח הראשי המקנים לו את אמינותו.
* התכנית הקיימת ברכיב ROM BIOS.

מכיון שקיים מבחר גדול של מחשבים תואמי יבמ, ניתן למצוא ביניהם גם כאלה שאיכותם וביצועיהם אינם ברמה של מחשבי יבמ מקוריים. באותה מידה, גם ישנם מחשבים תואמים, אשר עולים באיכותם וביצועיהם על מחשבי יבמ המקוריים.

כל העקרונות ומפרטי החומרה בספר מתייחסים גם למערכות PS/2 ו-PS/1 של יבמ. ההבדל העקרוני בין מערכות אלו לבין מערכות PC קיים רק בחיבור בין הערוצים לכרטיסי ההרחבה. במחשבים PS/2 משתמשים במיקרו ערוץ - Micro channel - ולא בערוץ BUS (ראה סעיף 2.7.5).

2.3 סוגי מעבדים ומהירויות עבודה

המעבד - CPU - הינו יחידת העיבוד המרכזית של מערכת המחשב האישי והוא המקנה לה את כוח החישוב ויכולת ביצוע הפקודות. סוג המעבד הקיים במערכת קובע את מהירות הביצוע של מערכת המחשב ותכונותיה. המעבד פועל במהירות, אשר נקבעת על ידי **גביש** (Crystal) המחובר אליו, ומכאן שהגביש קובע את מהירות הביצוע של הפקודות במחשב. שני מעבדים זהים יבצעו פעילות זהה במהירות שונה, אם תדר הגביש המחובר אליהם שונה. מעבדים שונים הפועלים בתדר גביש זהה יבצעו פעילות זהה במהירות שונה. הבדל זה במהירות נובע מארכיטקטורה פנימית שונה, אשר מקנה יכולת ביצוע שונה. הגביש נמצא על גבי הלוח הראשי בתוך תיבה מתכתית ותדר התנודות אותו הוא מספק למעבד רשום על גבה.

התדר של תנודות הגביש אינו בהכרח התדר שבו פועל המעבד במערכת. במערכות מחשבים אישיים המבוססות על מעבדי 8088/8086 יש לחלק את תדר הגביש ב-3, ורק אז נקבל את התדר שבו יפעל המעבד. במערכות מבוססות על מעבדי 80286/80386 יש לחלק את תדר הגביש ב-2 ואז מתקבל תדר העבודה של המעבד. במערכות מחשב המאפשרות לבחור בין שני תדרי פעילות שונים נמצא שני גבישים, אשר כל אחד מהם מספק תדר פעילות אחד בלבד. לדוגמה, במערכת מחשב אישית המבוססת על מעבד 8088 ומסוגלת לפעול בשתי מהירויות 10Mhz ו-4.77Mhz, נמצא שני גבישים, אשר מספקים את תדרי הפעילות למחשב: האחד בעל ערך של 30Mhz והשני בעל ערך של 14.31818Mhz. יש לזכור שאין משמעות הדבר שהמעבד פועל במהירות 30Mhz !.

את סדרת המעבדים שנוכל למצוא על גבי הלוח הראשי ניתן לסדר ברשימה הירכית על פי ביצועיהם ואפשרויותיהם. בראש הפירמידה יהיו בעלי הביצועים הגבוהים ביותר ובבסיסה - בעלי הביצועים הנמוכים ביותר והמעבד הישן ביותר. הנטיה בשוק המחשבים כיום היא לשלב את מעבד 80286 כתקן מינימלי למערכת המחשב. חברת יבמ, למשל, הפסיקה לפני מספר שנים את ייצור המחשבים האישיים המבוססים על מעבד 8088. כפי שכבר הוזכר, תכניות רבות, אשר מותאמות למעבדי 80286 ומעלה אינן ניתנות לביצוע על מעבדי 8088/8086. לדוגמה, רשימה של מחשבי יבמ:

80486	מעבד	PS/2 95 ,486
80386	מעבד	PS/2-80 ,386DX
80386SX	מעבד	386SX, PS/2-70
80286	מעבד	PS1, PS/2-50 ,60 PC-AT
8086	מעבד	PS/2-25,30
8088	מעבד	PC-XT

מהאמור לעיל נראה, שהשימוש במעבדי 8088 ילך וירד. אבל במצב הקיים, רוב המחשבים האישים הקיימים היום בארץ הם מסוג התואם XT שבו המעבד הוא מסוג 8088. מכיון שמחירי המערכות המשוכללות יותר יורדים, ומכיון שאורך חייו של מחשב אישי הוא לרוב 4 עד 10 שנים, סביר להניח שעל אף העובדה שהיום מעבדי 8088 הם הרוב, בעשור הבא לא נוכל למצוא אותם.

2.3.1 מחזורי המתנה

על גבי הלוח הראשי ישנם סוגים שונים של התקנים ורכיבים בעלי תכונות שונות. המעבד המרכזי פועל בצורה מחזורית ובכל מחזור הוא מבצע פעולה אחת. משך כל מחזור מכונה "קבוע מראש" ומורכב מארבעה חלקים, או **מחזורי שעון** (Clock Cycles). מחזור המכונה מגדיר את משך הזמן הנדרש להשלמת ביצוע פעולה אחת של המעבד מתחילתה ועד סופה. כל פנייה של המעבד לאחד ההתקנים, או הרכיבים, במערכת צריכה להסתיים בדיוק בתוך מחזור מכונה כזה. התקן או רכיב, אשר מאחר ואינו מסוגל להשלים את הפעילות בתוך מחזור המכונה, יפגע בביצוע של מחזור המכונה הבא ועלול לשבש ואף לפגוע בביצוע הפעילות במחשב.

אם כל ההתקנים השונים המחוברים למערכת המחשב היו מותאמים במהירותם למהירות הפעילות של המעבד הראשי (אשר התדר שלו נקבע על פי הגביש), כולם היו מסוגלים לסיים את הפעילות בתוך מחזור המכונה והמערכת היתה מתפקדת בצורה תקינה ויעילה. אך אם מהירות אחד ההתקנים או כולם קטנה יותר ממהירות המעבד, עלולה להיווצר בעיה בגישה אליהם.

ככל שעולים במהירות הביצוע של המעבד, מחיר הרכיבים המותאמים למהירות זאת מתייקר בהתאם. בשל מחיר זה, ובשל העובדה שקיימים התקנים אשר אינם מסוגלים לפעול במהירות בה פועל המעבד, נוצר צורך לאפשר למעבד להתקשר להתקנים איטיים יותר ממנו. המנגנון המאפשר למעבד להתקשר עם התקנים ורכיבים איטיים ממנו נקרא **מחזור המתנה** (Wait State). הוא מאפשר להתקנים האיטיים לסמן למעבד כי אינם מסוגלים לסיים את הפעולות בזמן, ויש להרחיב עבורם את משך זמן ביצוע מחזור המכונה. הרחבה של מחזור המכונה מאפשרת להם לסיים את פעילותם בתוך "מחזור המכונה" בצורה תקינה. הסימון של ההתקנים נקלט על ידי המעבד בזמן מחזור המכונה וכתוצא מכך הוא אינו משלים את המחזור בזמן "הקבוע מראש", אלא רק לאחר שההתקן מסמן כי סיים את פעילותו. מחזורי ההמתנה גוזלים זמן מהמעבד המרכזי ובכך מורידים את ביצועי המחשב.

במערכת מחשב שבה קיים **0 WAIT STATE** הכוונה היא, כי כל ההתקנים והרכיבים הנמצאים על גבי **הלוח הראשי** בלבד מותאמים למהירות העבודה של המעבד ואין צורך במחזורי המתנה לצורך פניה אליהם. ביצועים של מערכת מחשב זו מתאימים בצורה מושלמת לתדר שבו פועל המעבד, וזהו בעצם המצב

* מהירות הביצוע של המערכת.
* העיצוב החיצוני והמכני של מערכת המחשב.
* טיב הייצור וההרכבה של הלוח הראשי המקנים לו את אמינותו.
* התכנית הקיימת ברכיב ROM BIOS.

מכיון שקיים מבחר גדול של מחשבים תואמי יבמ, ניתן למצוא ביניהם גם כאלה שאיכותם וביצועיהם אינם ברמה של מחשבי יבמ מקוריים. באותה מידה, גם ישנם מחשבים תואמים, אשר עולים באיכותם וביצועיהם על מחשבי יבמ המקוריים.

כל העקרונות ומפרטי החומרה בספר מתייחסים גם למערכות PS/2 ו-PS/1 של יבמ. ההבדל העקרוני בין מערכות אלו לבין מערכות PC קיים רק בחיבור בין הערוצים לכרטיסי ההרחבה. במחשבים PS/2 משתמשים במיקרו ערוץ - Micro channel - ולא בערוץ BUS (ראה סעיף 2.7.5).

2.3 סוגי מעבדים ומהירויות עבודה

המעבד - CPU - הינו יחידת העיבוד המרכזית של מערכת המחשב האישי והוא המקנה לה את כוח החישוב ויכולת ביצוע הפקודות. סוג המעבד הקיים במערכת קובע את מהירות הביצוע של מערכת המחשב ותכונותיה. המעבד פועל במהירות, אשר נקבעת על ידי **גביש** (Crystal) המחובר אליו, ומכאן שהגביש קובע את מהירות הביצוע של הפקודות במחשב. שני מעבדים זהים יבצעו פעילות זהה במהירות שונה, אם תדר הגביש המחובר אליהם שונה. מעבדים שונים הפועלים בתדר גביש זהה יבצעו פעילות זהה במהירות שונה. הבדל זה במהירות נובע מארכיטקטורה פנימית שונה, אשר מקנה יכולת ביצוע שונה. הגביש נמצא על גבי הלוח הראשי בתוך תיבה מתכתית ותדר התנודות אותו הוא מספק למעבד רשום על גבה.

התדר של תנודות הגביש אינו בהכרח התדר שבו פועל המעבד במערכת. במערכות מחשבים אישיים המבוססות על מעבדי 8088/8086 יש לחלק את תדר הגביש ב-3, ורק אז נקבל את התדר שבו יפעל המעבד. במערכות מבוססות על מעבדי 80286/80386 יש לחלק את תדר הגביש ב-2 ואז מתקבל תדר העבודה של המעבד. במערכות מחשב המאפשרות לבחור בין שני תדרי פעילות שונים נמצא שני גבישים, אשר כל אחד מהם מספק תדר פעילות אחד בלבד. לדוגמה, במערכת מחשב אישית המבוססת על מעבד 8088 ומסוגלת לפעול בשתי מהירויות 10Mhz ו-4.77Mhz, נמצא שני גבישים, אשר מספקים את תדרי הפעילות למחשב: האחד בעל ערך של 30Mhz והשני בעל ערך של 14.31818Mhz. יש לזכור שאין משמעות הדבר שהמעבד פועל במהירות 30Mhz !.

את סדרת המעבדים שנוכל למצוא על גבי הלוח הראשי ניתן לסדר ברשימה הירכית על פי ביצועיהם ואפשרויותיהם. בראש הפירמידה יהיו בעלי הביצועים הגבוהים ביותר ובבסיסה - בעלי הביצועים הנמוכים ביותר והמעבד הישן ביותר. הנטיה בשוק המחשבים כיום היא לשלב את מעבד 80286 כתקן מינימלי למערכת המחשב. חברת יבמ, למשל, הפסיקה לפני מספר שנים את ייצור המחשבים האישיים המבוססים על מעבד 8088. כפי שכבר הוזכר, תכניות רבות, אשר מותאמות למעבדי 80286 ומעלה אינן ניתנות לביצוע על מעבדי 8088/8086. לדוגמה, רשימה של מחשבי יבמ:

הרצוי במערכת. בחלק מהמחשבים נמצאה אפשרות להתקין זיכרון איטי יותר במערכת מחשב מהירה ועל כן יש צורך במחזורי המתנה בכל גישה לזיכרון. מחשב XT סטנדרטי פועל במהירות 4.47Mhz ואינו דורש מחזורי המתנה כלשהם. מרבית התואמים מסוגלים לפעול במהירות 10/8Mhz על ידי מעבר למצב הנקרא Turbo. על כן נמצא שבחלק ממערכות אלו נדרשים מחזורי המתנה כאשר הם פועלים במצב זה: Wait State 1.

במערכות מחשב מתקדמות מסוג 80286 ומעלה הפועלים במהירות גבוהה (20-33Mhz) ההתקנים הפונים למעבד אינם פועלים במהירות שבה הוא פועל. על כן, בכל פניה של המעבד להתקנים אלה יופעל מנגנון מחזורי ההמתנה. מסיבה זו מחברים למחשב התקני **זיכרון מטמון** (Cache) קטנים ומהירים, כפי שמוסבר בהמשך. יש לזכור, כי מחזור מכונה מורכב מארבעה שלבים, כך שמחזור המתנה אחד נוסף יוריד מביצועי המערכת כ-20/25 אחוזים וזה בהחלט משמעותי. כלומר, מחשב של 10 MHz יתפקד כמחשב של 8MHz!

במערכות מהירות רכיבי זיכרון ROM שעל גבי הלוח הראשי ורכיבי זיכרון ROM הקיימים על גבי כרטיסי התיאום, הינם בדרך כלל איטיים יותר מרכיבי זיכרון RAM. לכן נמצא על גבי הלוח הראשי מעגל המבצע מחזורי המתנה בגישה לזיכרון זה.

2.3.2 הערוצים במעבד

מבנה החומרה של **הערוץ** (Bus) שדרכו מתקשר המעבד להתקנים המחוברים אליו משפיע על הקצב המירבי של העברת הנתונים בין התקנים הקשורים למחברי ההרחבה לבין המעבד. המבנה הפיסי גם יכול לגרום לשיבוש בהעברת הנתונים, כמו למשל המקרה שבו שני קווי נתונים שפרושים במקביל לאורך דרך ארוכה עלולים ליצור השראה הדדית בפעילות בתדר גבוה.

ישנם שלושה תקנים עיקריים למבנה הערוצים השונים במחשב האישי:

Isa Bus: הערוץ התקני במחשבי XT, AT, 386. במחשבי XT ערוץ הנתונים הוא ברוחב 8 סיביות, ב-AT וב-386SX - 16 סיביות וב-386 - 32 סיביות. הערוץ מאפשר העברת נתונים בקצבים של 8MHz עד 12MHz. מכאן, שבכל מערכת מחשב שבה המעבד פועל בתדר גבוה יותר מתדר הערוץ, חייבים להשתמש במנגנון מחזורי ההמתנה בכל גישה של המעבד להתקן כלשהו דרך הערוץ הזה.

Eisa Bus: תקן הערוצים הפועלים בקצב 20Mhz ונמצאים במערכות 286 ו-386 מהירות. ערוץ זה מאפשר להעביר נתונים בקצבים גבוהים יותר מאשר ערוץ Isa. חסרונו הגדול בכך שכרטיסי מתאם מסך אינם מסוגלים לפעול בקצב זה ולכן מהירות העבודה הגרפית אינה גדלה כאשר מחברים מתאם מסך לערוץ זה.

מיקרו ערוץ (Micro Channel): מעביר נתונים בין המעבד לבין התקנים המחוברים למיקרו ערוץ בקצב עד 14Mhz. ערוץ זה נמצא במערכות PS/2 של יבמ. תקן מיקרו-ערוץ אינו מקובל עדיין כסטנדרד ומספר המערכות הפועלות בתקן זה מלבד PS/2, קטן יחסית.

2.4 מערכת הזיכרון במחשב

על גבי הלוח הראשי במחשב האישי קיימים שני סוגי זיכרון עיקריים:

2.4.1 זיכרון ROM/EPROM (התקני ROM BIOS)

זיכרון ROM מותקן בלוח הראשי של כל מחשב אישי וכשמו, הוא מאפשר קריאה בלבד. תוכן זיכרון ROM נכתב על ידי יצרן המחשב בבית החרושת ומכיל את **תכנית ההפעלה הראשונית** של המחשב האישי הקרויה ROM BIOS. הוא משמש למחשב האישי לפעולת **האתחול** (BOOT) ולהפעלת התקני חומרה וכרטיסי תיאום המחוברים אליו. זיכרון ROM/EPROM אינו נמחק עם כיבוי המחשב, בניגוד לזיכרון RAM. הוא קטן יחסית לזיכרון RAM וגודלו בדרך כלל 8KByte במחשב המבוסס על מעבדי 8088/8086 ו-16KByte עד 128KByte במחשבים המבוססים על מעבדי 80286 ומעלה.

כאשר מדליקים את המחשב, או מבצעים פעולת Reset למערכת, פונה המעבד המרכזי אל רכיב ROM BIOS ומבצע את **תכנית האתחול** הכתובה בו. תכנית האתחול בודקת את מערכות החומרה המותקנות על הלוח הראשי, או מותקנות בו באמצעות מחברי ההרחבה ומתריעה על תקלות כלשהן במערכת. בגמר הבדיקה מספקת תוכנה זו כלים בסיסיים להפעלת ההתקנים הקשורים למערכת (מסך, מקלדת, מדפסת, דיסקים וכו').

במחשבי AT ומעלה ניתן לעתים למצוא בתכנית BIOS גם תכניות מתוחכמות יותר המבצעות Disk Cache, טעינת חלק מה-ROM ל-RAM, אתחול לדיסקים קשיחים ועוד. בסוף תהליך האתחול דואגת התכנית הנמצאת ב-ROM BIOS לטעון את רשומת BOOT מדיסקט, או דיסק, ולהעביר אליה שליטה לצורך המשך טעינת מערכת ההפעלה (על תהליך האתחול - BOOT - ראה בפרק על DOS).

על הלוח הראשי, כפי שניתן לראות בטבלת מיפוי זיכרון, ניתן למצוא מקום השמור לרכיבי זיכרון נוספים המאפשרים קריאה בלבד. רכיבים אלה הינם אופציונליים ולא תמיד קיימים במערכת.

תכנית ROM BIOS בודקת אם קיימים רכיבי ROM נוספים במערכת ועל כן אפשר לחבר למערכת המחשב האישי כרטיסים מתקדמים נוספים, אשר תכנית ROM BIOS אינה מכירה. הוספת רכיבי ROM/EPROM מאפשרת ליצרני כרטיסי התיאום לשים את תכנית ההפעלה של הכרטיס המסוים שהם מספקים על גבי הכרטיס עצמו. תכנית ROM BIOS מבחינה ברכיבים אלה ומבצעת את התכנית הכתובה בהם. התקנים אלה מחייבים אתחול ותכנות לפני שניתן להפעיל אותם.

כדי שתכנית ROM BIOS תזהה רכיב ROM הנמצא על גבי כרטיס חייבים לציין לה כי קיים רכיב כזה. לכן, בתחילת הנתונים הכתובים ב-ROM נמצא תמיד קוד ייחודי (המלה AA55Hex תופיע בתחילת כל 2KByte בזיכרון ROM) אשר מציינת לתוכנת ROM BIOS כי קיים רכיב ROM ויש לבצע את התכנית הכתובה בו בזמן תהליך האתחול של המחשב. כאשר תכנית ROM BIOS מזהה התקן ROM נוסף, היא מעבירה שליטה לתכנית הכתובה בו. תכנית ROM הנוסף מבצעת את תכנית האתחול עבור כרטיס התיאום ובגמר האתחול היא מחזירה שליטה לתכנית ROM BIOS בכדי לאפשר סיום תקין של תהליך BOOT. שימוש ברכיבי

ROM להפעלת התקנים חוסכת את הצורך להוסיף **תכנית הפעלה להתקן** (Device Driver) בקובץ CONFIG.SYS של מערכת ההפעלה DOS, אשר תופסת מקום יקר מזיכרון RAM במערכת המחשב. רכיבי ROM קיימים לדיסק הקשיח במערכות XT, לכרטיסי רשתות תקשורת, למתאמי מסכים מתקדמים (EGA ומעלה) וכו'.

במחשבים מהירים (80286 ומעלה) רכיבים אלה אינם פועלים במהירות שבה פועל המעבד. לכן, כל גישה אליהם לצורכי ביצוע תכניות BIOS תגרום **למחזורי המתנה** (Wait States) ותאיט את קצב פעולת המחשב (ראה סעיף על Shadow Ram).

מספר פסיקת שירות	תיאור תכנית השירות
5 Hex	הדפסת עותק של המסך על גבי המדפסת.
10 Hex	מסך - תכניות גישה למסך (טקסט וגרפיקה).
11 Hex	פירוט ציוד חומרה המחובר למערכת המחשב.
12 Hex	כמות זיכרון - כמה זיכרון קיים במערכת.
13 Hex	דיסק / דיסקט - קריאה, כתיבה, גישה, פורמט.
14 Hex	תקשורת טורית - פרמטרים, שידור, קליטה.
15 Hex	טיפול בזיכרון הרחבה Extended לצורכי קריאה וכתיבה. טיפול בקלטות, טיפול ב-Joy-Stick, גודל זיכרון מעל 1MByte, במחשבי AT ומעלה, מעבר למצב מוגן (Protected) וחזרה.
16 Hex	לוח מקשים - מצב, קריאה.
17 Hex	מדפסת - כתיבה, אתחול, מצב מדפסת.
19 Hex	BOOT מחדש.
1A Hex	שעון זמן אמת ותוכנה - כיוון, קריאה, כיוון אזעקה בשעון זמן אמת.
1B Hex	טיפול במקש Control Break.
1C Hex	שעון זמן - אפשרות "לשתול" תכנית משתמש שתתבצע 18.2 פעם בשנייה.
1D Hex	אתחול מתאם מסך ובירור פרמטרים של מסך.
1E Hex	פרמטרים של כונן דיסקטים.
4A Hex	שעון זמן אמת - אפשרות "לשתול" תכנית שתתבצע לאחר פרק זמן שניתן לתכנות (שעון מעורר).

טבלת פסיקות שירות הנמצאות ב-ROM BIOS

2.4.2 זיכרון RAM ראשי

זיכרון RAM משמש את המחשב האישי בפעילות השוטפת. הזיכרון הראשי מאפשר למעבד פעולות קריאה וכתיבה תוך כדי פעילות המערכת. תכנית המתבצעת במחשב חייבת להימצא בזיכרון הראשי על מנת שניתן יהיה לבצע אותה. התקשורת בין המעבד להתקנים ההיקפיים מתבצעת בחלקה באמצעות איזורים בזיכרון הראשי. זיכרון RAM נדיף, כלומר בכיבוי המתח הוא נמחק והתוכן שלו אובד.

זיכרון RAM שעל הלוח הראשי נקרא **RAM דינמי** (DRam). התכונה המאפיינת RAM מסוג זה, היא הצורך לבצע כל הזמן **רענון** (Refreshing) של תוכן הזיכרון. אם לא יתבצע רענון, תוכן הזיכרון יימחק לאחר פרק זמן קצר ביותר (מספר מילי-שניות). הרענון נעשה על ידי קריאות יזומות של החומרה מזיכרון RAM בכל פרק זמן קצוב. מכיון שהפעולה נעשית בחומרה המשתמש אינו מעורב בה. עם זאת, ברוב המערכות מנגנון הרענון גוזל זמן יקר מזמן הביצוע של המעבד ומוריד את ביצועי המערכת.

תכונות אלו של RAM אינן מאפשרות לשמור בו נתונים קבועים, אלא רק נתונים זמניים המשמשים את המעבד לצורכי פעילות שוטפת בלבד. בתהליך האתחול טוענת מערכת ההפעלה אל זיכרון RAM את תכנית מערכת ההפעלה DOS. היא משתמשת אך ורק ב-640KByte הראשונים במרחב הזיכרון, למרות שמרחב הזיכרון בפועל גדול יותר (ראה "טבלת מיפוי זיכרון"). בזמן ביצוע תכניות משתנה תוכן הזיכרון בצורה דינמית על פי הפעולות שאנו מבצעים במחשב.

ארגון הזיכרון ומנגנון בקרת Parity

רכיבי הזיכרון נמצאים על גבי הלוח הראשי במחשב. במקרים מסוימים נמצא כי זיכרון RAM נמצא על כרטיס תיאום המשמש כהרחבת זיכרון. רוב המחשבים משתמשים ברכיבי זיכרון ובתצורת זיכרון DRam תקניים. במחשבים אלה ניתן לזהות את רכיבי הזיכרון לפי שורות של 9 רכיבים בכל שורה. כזכור, רוחב הזיכרון במחשב הינו בית (Byte) כלומר, 8 סיביות. כל רכיב זיכרון הוא ברוחב של סיבית אחת וכך, שמונה רכיבים בשורה מרכיבים ביחד בית אחד. הרכיב התשיעי בכל שורת זיכרון משמש את המחשב לצורכי בדיקת תקינות של הזיכרון ונקרא **סיבית בדיקת זוגיות** (Parity).

כאשר נרשם נתון לזיכרון מחושבת סיבית הזוגיות (התשיעית) על פי תוכן הבית שנרשם בו. כאשר מבקשים לקרוא בית מהזיכרון מחושבת סיבית הזוגיות על פי תוכן הבית הנקרא ממנו. הערך (0 או 1) של הסיבית המחושבת מושווה לערך של הסיבית התשיעית של תא הזיכרון שנקרא. אם הערך של שתי הסיביות שווה, סביר שהזיכרון תקין ואם ערכן שונה - ישנה תקלה בזיכרון והוא אינו ראוי לשימוש. כאשר המחשב מזהה תקלת זיכרון הוא מודיע על שגיאה ומפסיק פעילות.

בזמן האתחול בודקת תכנית האתחול הנמצאת ב-ROM BIOS את זיכרון RAM. אם היא מזהה רכיב זיכרון שאינו תקין, היא מציגה הודעה מתאימה על המסך. במערכות יבמ ישנות תופיע הודעת שגיאה והמחשב יעצור את תהליך האתחול. במערכות חדישות יותר, שבהן המחשב מציג על המסך את ביצוע הבדיקה, הוא לא יכלול בספירת כתובות הזיכרון תאי זיכרון שאינם תקינים. כך, השוואה בין כמות הזיכרון שמצא המחשב לבין כמות הזיכרון שקיימת פיסית תיתן מידע על תקינות הזיכרון.

רכיבי הזיכרון

במחשבים המבוססים על מעבדי 8088/8086 ישנה כיום אפשרות להתקין על הלוח הראשי זיכרון RAM בגודל עד 1MByte. במחשבים ישנים יותר, ניתן היה להרכיב על גבי הלוח הראשי עד 256KByte, אך במרביתם ניתן להרחיב את הזיכרון הראשי עד ל-640KByte על גבי הלוח הראשי. במחשבי AT (286)

ניתן כיום להתקין על גבי הלוח הראשי במחשבי 386 4MByte מרכיבים בדרך כלל על הלוח הראשי 2MB או 4MB ולפעמים אפילו 16MB זיכרון ראשי.

רכיבי זיכרון הקיימים בשוק שונים זה מזה בקיבולת הזיכרון שלהם וברוחב הזיכרון. בספר ההפעלה של כל מחשב רשום אילו רכיבי זיכרון DRam יש לחבר לו, על פי גודל זיכרון הנדרש. בנוסף, רשום איזה מתגים או מגשרים (jumpers) יש לחבר, או לשנות, על מנת להודיע למחשב כמה זיכרון מותקן על הלוח הראשי ובאיזו תצורה. להלן מספר דוגמאות אופייניות לרכיבי זיכרון נפוצים הנמצאים בשימוש במחשבים השונים:

רותב * אורך	הסבר	דוגמה לרכיב
64K*1Bit	זיכרון של 64K תאים כשכל תא ברוחב של סיבית אחת.	4164
64K*4Bit	זיכרון של 64K תאים כשכל תא ברוחב של ארבע סיביות.	4464
256K*1Bit	זיכרון של 256K תאים כשכל תא ברוחב של סיבית אחת	514256, 33256 41256
1024K*1Bit	זיכרון של 1M תאים כשכל תא ברוחב של סיבית אחת.	41024

השפעת התקדמות הטכנולוגיה על רכיבי הזיכרון

טכנולוגיית VLSI, אשר מהווה את הבסיס לפיתוח מרבית רכיבי המחשב וביניהם גם רכיבי הזיכרון, נמצאת בתהליך מהיר של התפתחות. התפתחות טכנולוגית משמעותה מזעור של הגודל הפיסי של הרכיבים בצד קיבולת הולכת ועולה של נתונים שניתן לאגור בהם. המזעור מאפשר לדחוס יותר חלקים לתוך מארז אחד. מכאן, שיש יותר זיכרון בתוך "ג'וק" אחד. רכיב 41024 למשל, מכיל 1024Kbit שהם 128Kbyte. ניתן למצוא היום רכיבי זיכרון זעירים יותר המכילים 4MByte. התחזית היא שקיבולת זו תלך ותגדל.

המזעור מקטין את גודל המרכיבים של הרכיב וגם את אורך החיבורים שביניהם. מרחק קטן יותר בין החלקים השונים ברכיב מקטין את הקיבוליות וההשראות של החיבורים ומאפשר הגדלת מהירות הפעולה של הרכיב והקטנת ההספק הנדרש להפעלתו. כל הרכיבים במחשב האישי זקוקים למתח על מנת שיפעלו. מתח זה קרוי מתח TTL (מתח של 5V). המהירות המירבית על פי שיטת עבודה זו כיום הינה היא בערך 50Mhz וצפוי שהיא תלך ותגדל.

מחשבים מרכזיים (Main Frames) פועלים בשיטת ECL, שבהם ניתן להגיע למהירות מירבית של 500Mhz.

2.4.3 אופני גישה לזיכרון DRam

ארגון הזיכרון במחשב נקבע על ידי היצרן, ולמשתמש אין אפשרות לשנות צורת ארגון זו. צורת ארגון הזיכרון "שקופה" לתכניות מערכת ההפעלה ולתכניות היישום השונות.

שיטה תקנית בסיסית – Basic Row/Column Method

לזיכרון DRam קיים מעגל רענון. פעולת הרענון משתמשת בחלק מזמן המחזור של המעבד וגורמת לכך שהמערכת תפעל בקצב איטי יותר ממערכת זהה, אשר אינה זקוקה למעגל רענון (חיבור של זיכרון RAM סטטי למשל). גישה לכל תא זיכרון DRam נעשית באמצעות כתובת המורכבת משני חלקים: **כתובת שורה** (Row Address) **וכתובת עמודה** (Column Address). כתובת זו מגדירה תא זיכרון יחיד - **כתובת זיכרון** (Memory Address). באופן זה ניתן לראות את הזיכרון כמטריצה של תאים כשחיתוך של כתובת שורה וכתובת עמודה מגדיר את הכתובת של כל תא בזיכרון.

שיטת עמודה סטטית – Static Column Method

שיטה זו מיועדת למערכות 80286 ומעלה. הזיכרון מאורגן באופן דומה לארגון הבסיסי של שורה/עמודה. שיטה זו יעילה כאשר יש גישות עוקבות לזיכרון DRam אשר כתובת העמודה המזהה אותן (Column Address) זהה. כאשר ניגשים ברצף למספר תאים בעלי כתובת עמודה משותפת, ניתן לחסוך את כתובת העמודה ועל ידי כך לקצר את זמן הגישה לזיכרון.

זיכרון עם רווח ביניים – InterLeaved Memory

שיטה זו מיועדת למערכות 80386 ומעלה. הזיכרון מאורגן בשתיים, או בארבע מחיצות. כאשר המעבד מבצע פעולה במחיצה אחת של הזיכרון, מנגנון הרענון פועל במקביל על מחיצה אחרת. כאשר הזיכרון מאורגן עם רווח ביניים, זמן המחזור כולו מוקצב למעבד ומנגנון הרענון אינו גוזל מזמן זה. כתוצאה – ביצועי מערכת המחשב עולים.

ארגון בדפים – Paged Mode

שיטה זו מיועדת למערכות 80386 ומעלה. הזיכרון מאורגן **בדפים** (Pages) וכך גם הגישה אל הנתונים הנמצאים בו, נעשית לפי דפים. כל הגישות הרצופות לתאי זיכרון הנמצאים באותו דף ייעשו ללא מחזורי המתנה. בשיטת ארגון זו ניתן לשפר את ביצועי המערכת מכיון שמרבית הגישות של המעבד לזיכרון הן עוקבות (עיקרון המקומיות). גם נמצא, כי הפעולות המתבצעות בין הרכיבים שעל גבי הלוח הראשי נעשות כמעט ללא מחזורי המתנה. ביצועי המערכת טובים יותר ממערכת ללא ארגון דפים.

כל השיטות המתוחכמות לגישה לזיכרון קיימות רק במחשבים מהירים, אשר תדרי הפעולה שלהם מעל 20MHz. במחשבים אלה פועל זיכרון DRam במהירות נמוכה יותר ממהירות הפעולה של המעבד, ולכן הדבר מחייב מחזורי המתנה. שיטות ארגון אלו נועדו לחסוך בזמן הגישה, למנוע מחזורי המתנה ובכך לשפר את ביצועי המערכת. מהירות הפעולה האיטית של זיכרון DRam באופן יחסי למהירות המעבד הביאה להוספת **זיכרון מטמון** (Cache), כפי שנראה בהמשך. זיכרון המטמון מהיר יותר מזיכרון DRam ומאפשר למעבד להתקשר אליו במהירות גבוהה יותר מאשר זיכרון DRam מסוגל לפעול.

2.4.4 מרחב כתובות הזיכרון

מרחב הזיכרון המירבי הניתן לגישה על ידי מעבדי 8088/8086 הוא 1MByte. במרחב כתובת זה קיים ייעוד לאיזורים שונים, אשר משותף לכל מערכות יבמ והתואמים. לצורך תאימות תוכנה/חומרה כלפי מטה במערכות המחשב, גם המערכות המתקדמות יותר ממערכות XT, אשר מסוגלות לגשת למרחב כתובות גדול בהרבה מ-1MByte, מייעדות את 1MByte הראשונים של זיכרון DRam לאותו הייעוד הקיים במערכות XT. מרחב הזיכרון עד 1MByte מחולק בחומרה לשני איזורים:

* **640K - 0: מרחב התוכנה** – מרחב זיכרון רציף המיועד לזיכרון DRam ומשמש לתכניות שונות במערכת.
* **1M - 640K: מרחב החומרה** – מרחב זיכרון לא רציף, המשמש את כרטיסי התיאום ומעגלי החומרה של התקנים שונים המתחברים למערכת. לא כל מרחב הזיכרון נמצא בשימוש.

2.4.4.1 מרחב הכתובות עד 640Kbyte (מרחב התוכנה)

לזיכרון DRam מוקצה מרחב הכתובות 00000-9FFFFHex (0-640KByte). מרחב כתובות זה מחולק למספר איזורים מוגדרים מראש, המשמשים את מערכת המחשב בפעילויות שונות המתבצעות בפיקוח מערכת ההפעלה DOS:

00000-003FF Hex

מרחב זה הוא 1KByte התחתונים של הזיכרון אשר בו נמצאת טבלת הווקטורים של פסיקות המערכת. הטבלה כוללת רשימת מצביעים לכתובת של תכניות השירות של ROM BIOS, מערכת ההפעלה, בייסיק, רשתות תקשורת, כרטיסי חומרה שונים ועוד.

כל שירותי מערכת ההפעלה DOS ניתנים באמצעות שימוש במנגנוני הפסיקה, אשר מאפשרים שקיפות של הכתובות של תכניות השירות. כלומר, הם פוטרים את המתכנת מהצורך לדעת את הכתובת האמיתית של כל תכנית שירות אשר עלולה להשתנות בין יצרני חומרה שונים, או בין גירסאות שונות של מערכת ההפעלה. תכנית האתחול של מערכת המחשב האישי שנמצאת ב-ROM BIOS ותכנית האתחול של מערכת ההפעלה DOS מעדכנות את טבלת וקטורי הפסיקות. הן כותבות בכל כניסה בטבלה את הכתובת של תכנית השירות המתאימה לכניסה זו. כאשר הטבלה מעודכנת בכתובות תכניות השירות, יכול המחשב לפעול בצורה תקינה.

00400-004FF Hex

באיזור זיכרון זה שומרת תכנית ROM BIOS נתונים שונים הדרושים לה לצורך הפעילות המתבצעת במערכת המחשב. בין הנתונים הנשמרים באיזור זה נמצא: מקשים שנלחצו בלוח המקשים, תצורת חומרה של מערכת המחשב, כתובת כרטיס ההרחבה של המדפסת, כתובות כרטיסי תקשורת ועוד.

0500-06FF Hex

באיזור זה שומרת מערכת ההפעלה נתונים שונים הדרושים לה לצורך פעילות מערכת ההפעלה. שאר זיכרון DRam משמש את מערכת המחשב האישי לצרכים שונים, כפי שנפרט להלן.

בתהליך האתחול (BOOT) טוענת מערכת ההפעלה DOS לזיכרון DRam את הקבצים הבאים, זה אחר זה:

* **קובץ** IBMBIO.COM במערכת הפעלה IBM-DOS, או **קובץ** IO.SYS במערכת הפעלה MS-DOS.

* **קובץ** IBMDOS.COM במערכת הפעלה IBM-DOS, או **קובץ** MSDOS.SYS במערכת הפעלה MS-DOS.

* **תכניות להפעלת התקני חומרה שונים** (Device Drivers) על פי הסדר שבו הם מופיעים בקובץ CONFIG.SYS, אשר מורה למערכת ההפעלה לטעון אותם לזיכרון (לדוגמה DRIVER.SYS, DISPLAY.SYS, HIMES.SYS ועוד).

* **תכניות המגדירות את תצורת העבודה** של מערכת ההפעלה DOS. גם תכניות אלה נטענות על פי סדר הופעתן בקובץ CONFIG.SYS (לדוגמה: BUFFERS, FILES ועוד).

* **מעבד הפקודות** COMMAND.COM ותכניות הנטענות כתכניות שוכנות – TSR (resident - תכניות אשר נמצאות דרך קבע בזיכרון המחשב).

* **תכניות משתמש** שונות. בזיכרון DRam שנותר, משתמשת מערכת ההפעלה כדי לטעון תכניות יישומיות שהמשתמש מעוניין לבצע. תכניות המשתמש נטענות לזיכרון Dram הפנוי באמצעות תכנית שירות של מערכת ההפעלה, ובגמר הטעינה מתחיל הביצוע של התכנית האופן אוטומטי. בסיום התכנית מוחזרת השליטה למעבד הפקודות והאיזור בזיכרון, שנתפס על ידי תכנית המשתמש שהסתיימה, מתפנה לצורך טעינת תכניות משתמש אחרות.

את פירוט תפוסת זיכרון DRam במהלך העבודה במחשב ניתן להציג על גבי המסך (בגירסה 4.00 ומעלה של DOS) על ידי הקשת פקודת מערכת ההפעלה MEM/PROGRAM. פירוט תפוסת הזיכרון מאפשר למשתמש לברר כמה זיכרון קיים במערכת, כמה מתוכו פנוי להרצת תכניות ומה הם השימושים של מערכת ההפעלה בזיכרון שאינו פנוי למשתמש. קיימים בשוק כלי תוכנה שונים, כמו SysInfo של Norton, או של PCTools, אשר מאפשרים להציג פירוט זה בצורה בהירה ומובנת למשתמש.

גודל זיכרון DRam במחשב מוגבל וכל תכנית שנמצאת דרך קבע בזיכרון מקטינה אותו יותר ויותר. מכיון שכך, גודל תכנית מירבי שניתן לבצע במחשב תלוי בזיכרון הפיסי הקיים במערכת המחשב (לכל מערכת מחשב גודל שונה עד ל-640KByte), בגודל הזיכרון שבשימוש מערכת ההפעלה, סוג תכניות Resident שהופעלו, מספר Device Drivers שהופעלו ועוד. גירסאות שונות של מערכת ההפעלה DOS צורכות שטחי זיכרון בגודל שונה. ככל שגירסת מערכת ההפעלה עולה, מערכת ההפעלה תשתמש בזיכרון רב יותר. מצב זה גורם להקטנה של כמות הזיכרון הפנויה לשימושן של תכניות היישומים.

בגירסה 5 של DOS חלה פריצת דרך בכך שהיא מאפשרת לטעון חלקים ממערכת ההפעלה בזיכרון ההרחבה ועל ידי כך מאפשרת זיכרון פנוי רב יותר מתחת ל-640K עבור תכניות המשתמש.

2.4.4.2 מרחב הכתובות מעבר ל-640K (מרחב החומרה)

מרחב הזיכרון בכתובות 640K עד 1024K ‏(A0000-FFFFFHex) מחולק לאיזורים ייעודיים שונים, אשר מפורטים בטבלת מיפוי הזיכרון. חלקו העליון של מרחב הזיכרון מוקצה לרכיב ROM BIOS, מכיון שיצרן המעבד (חברת Intel) קבע, שלאחר פעולת Reset של מערכת המחשב (או הדלקה) מתחיל המעבד את ביצוע הפעולות מכתובת FFFF:0. על כן יש במרחב כתובות זה רכיב זיכרון מסוג ROM/EPROM המכיל תכנית, אשר אינה נמחקת בכיבוי המתח למערכת.

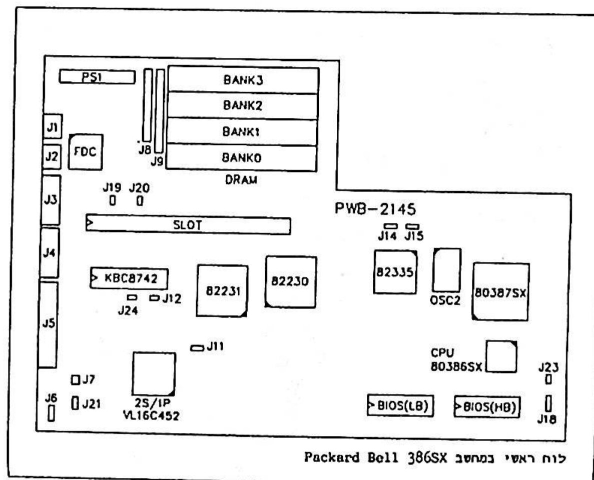
למרחב הזיכרון בכתובות C0000-CFFFFHex תפקיד מיוחד בתהליך האתחול (BOOT). תכנית האתחול מחפשת באיזור זה תכניות אתחול לרכיבים ולהתקנים שונים (ראה סעיף ROM BIOS למעלה).

מרחב הזיכרון בכתובות A0000-BFFFFHex מוקצה לכרטיסי התיאום למסכים השונים. לכל סוג של מתאם מסך מוקצה מרחב זיכרון בכתובת שונה ובקיבולת שונה בהתאם לנדרש מתכונות המסך. מכיון שכל כרטיס תיאום למסך ממופה במרחב כתובות שונה, ניתן לחבר בו-זמנית לאותו המחשב מתאמי מסך שונים.

כרטיסי חומרה שונים משתמשים במרחב הזיכרון הפנוי על פי צורכי הכרטיס מבלי להתנגש עם כתובות של כרטיסים אחרים.

2.4.4.3 מיפוי כתובות הזיכרון

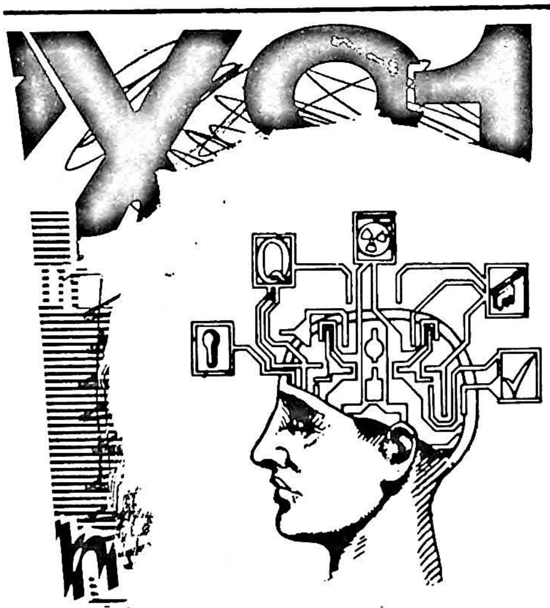
הטבלאות הבאות מפרטות את ייעוד מרחבי הכתובות השונים במחשב האישי, אשר נקראות **מפת כתובות** (Address map). כל שורה בטבלה מתארת איזור במרחב הכתובות הכללי ואת ייעודו של אותו איזור. האיזורים במרחב הזיכרון המוגדרים בטבלה כפנויים, מיועדים לכרטיסי תיאום נוספים. כאשר רוכשים כרטיסים אלה (כמו למשל, זיכרון הרחבה Expanded, או כרטיס תקשורת), היצרן רושם ומתעד את הכתובות שבשימוש הכרטיס. במרבית כרטיסי התיאום קיימת אפשרות לשנות את כתובת הכרטיס שנקבעה כברירת מחדל על ידי היצרן לכתובת פנויה במחשב האישי, על מנת שלא לחפוף עם כתובות של כרטיסים אחרים.



לוח ראשי במחשב Packard Bell 386SX

הכתובת		
ייצוג עשרוני	ייצוג הקסה	ייעוד איזור הזיכרון
0-1K	00000-003FF	טבלת וקטורי פסיקות חומרה ותוכנה במערכת.
1-640K	00000-9FFFF	מרחב עבודה עבור DOS.
640-704K	A0000-AFFFF	מתאם מסך VGA, EGA.
704-736K	B0000-B7FFF	מתאם מסך מונוכרום.
736-768K	B8000-BFFFF	מתאם מסך צבעוני CGA.
768-896K	C0000-DFFFF	זיכרון ROM/EPROM לצורכי הרחבת תוכנה ובקרת התקנים, הרחבת זיכרון Expanded, בקר דיסק, מתאם EGA וכו'.
896-960K	E0000-EFFFF	שמור על גבי הלוח הראשי.
960-1024K	F0000-FFFFF כולל ROM BASIC	ROM BIOS בדיקות, אתחול ותכנית הפעלה ראשונית. 8KByte האחרונים בלבד.

מיפוי כתובות זיכרון במחשבים עם מעבד 8088/8086



הכתובת ייצוג עשרוני	ייצוג הקסה	ייעוד איזור הזיכרון
0-1K	00000-003FF	טבלת וקטורי פסיקות חומרה ותוכנה.
1-640K	00000-9FFFF	מרחב עבודה עבור DOS.
640-704K	A0000-AFFFF	מתאם מסך VGA, EGA.
704-736K	B0000-B7FFF	מתאם מסך מונוכרום.
736-768K	B8000-BFFFF	מתאם מסך צבעוני CGA.
768-896K	C0000-DFFFF	זיכרון ROM/EPROM המשמש להרחבת תוכנה ובקרת התקנים, הרחבת זיכרון Expanded, מתאם EGA וכו'.
896-960K	E0000-EFFFF	רכיבי ROM BIOS או התקנים אחרים. מוכפל בכתובות FE0000-FEFFFF.
960-1024K גודל	F0000-FFFFF 16KByte-64KByte	ROM BIOS בדיקות אתחול ותכנית הפעלה ראשונית. מוכפל בכתובות FF0000-FFFFFF.
1M-16MBYTE	100000-FDFFFF	זיכרון RAM לשימוש כזיכרון Extended /Expanded.
	FE0000-FFFFFF	זיכרון לקריאה בלבד לצורכי הרחבת תוכנה ובקרת התקנים, בדיקות ואתחול. מוכפל בכתובות 0E0000-0FFFFF

מיפוי זיכרון במחשבי AT ומעלה

2.5 זיכרונות הרחבה Extended / Expanded

לפני שהינך ממשיך לקרוא, שים לב: אין לבלבל בין "זיכרונות הרחבה" לבין הגדלת (או "הרחבת") הזיכרון הבסיסי עד 640K. זיכרון המחשב עד 640KByte-ל נחשב **לזיכרון בסיסי**, אשר כל הגירסאות השונות של מערכת ההפעלה DOS יודעות לנצל.

מדוע זיכרון הרחבה?

מערכת ההפעלה DOS אינה "יודעת" לנצל את זיכרון המחשב שמעבר לזיכרון הבסיסי (מחסום 640KByte הראשונים) והדבר מהווה מכשול חמור. מגבלה זו מונעת מתכניות שונות את האפשרות להתפתח מבחינת גודלן, מורכבותן וכמות הנתונים שהן מסוגלות לגשת אליהן. המגבלה חמורה על רקע הגידול שחל בדרישות לכושר עיבוד ולזיכרון פנוי ליישומים, כתוצאה מן הגידול העצום שחל במהירות הביצוע של מחשבים אלה ובמרחב הזיכרון שניתן להתקין בהם.

במחשבים המבוססים על מעבדי 80286 ומעלה, קיימת אפשרות לפנות למרחב זיכרון גדול יותר מאשר מערכת ההפעלה DOS יכולה לנצל. למשל, מערכת ההפעלה DOS מפעילה את מעבד 80286 בצורת עבודה REAL MODE התואמת למעבד 8088/8088, ואינה מאפשרת לגשת למרחב זיכרון מעבר ל-1MByte. מגבלה זו הובילה לפיתוח פתרונות חומרה שונים, שנועדו לשבור את מחסום 640KByte בכל המחשבים האישיים, גם באלה המבוססים על מעבדי 8088/8086, ולאפשר גישה לתכניות למרחב זיכרון אשר גדול בהרבה מ-640KByte.

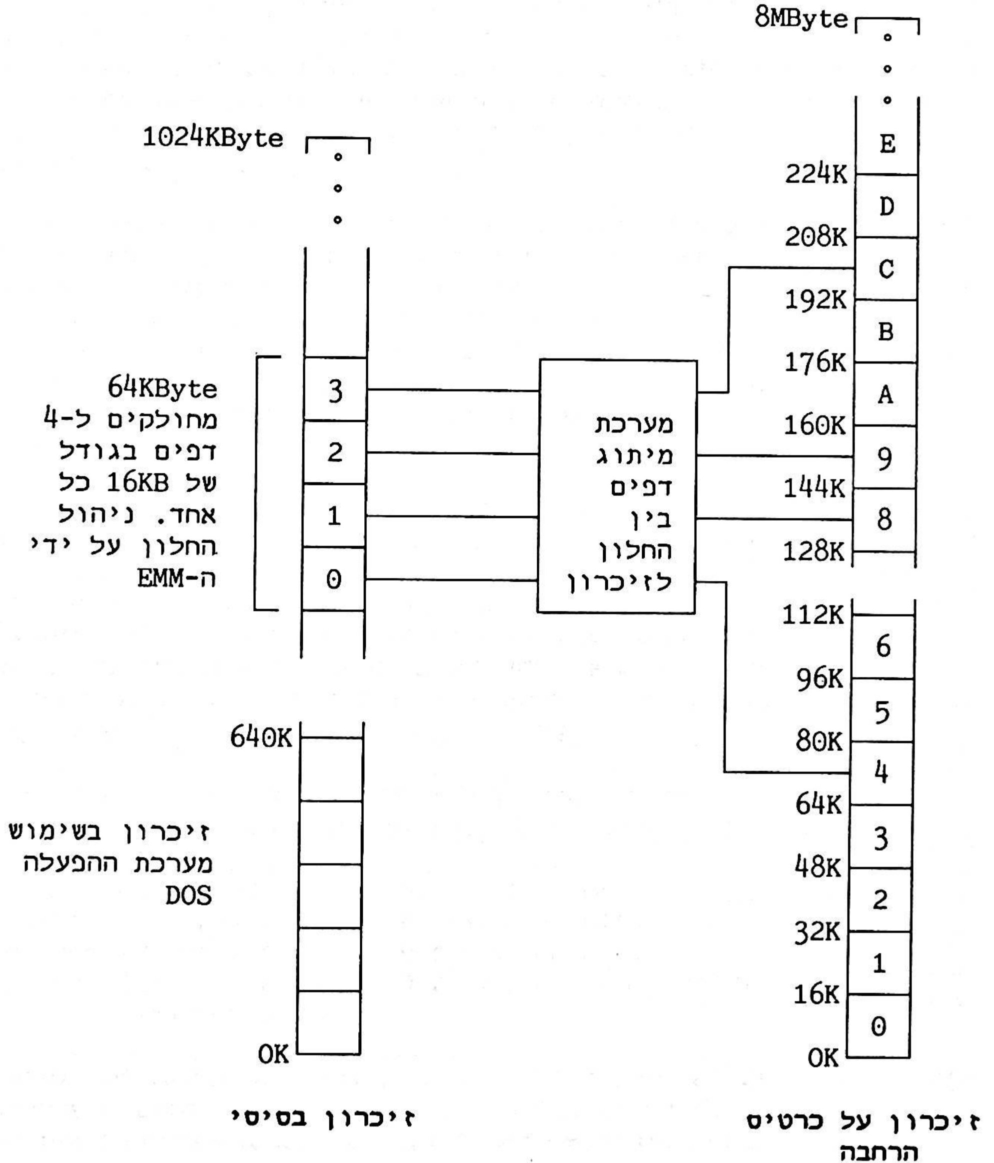
מעבדי 8088/8086 אינם מסוגלים לפנות לזיכרון גדול מ-1024K. הזיכרון עד 1024K מוקצה (כפי שמוסבר בסעיף **2.4.4**) כמעט כולו לשימושי תוכנה מלבד האיזורים המוקצים לחומרה. הצורך בניצול מרחב זיכרון גדול יותר הביא לחיפוש פתרון של מעגל חומרה, אשר יציג בפני התכניות השונות מרחב זיכרון גדול בהרבה ממרחב זיכרון החומרה הפנוי עבורו במרחב הכתובות של המעבד. מכיון שהמעבד יכול לגשת בו-זמנית רק למיליון כתובות שונות, הזיכרון הגדול מוצג לתכנית בחלקים ולא בו-זמנית.

מעבדי 80286 פועלים ב-Real Mode תחת מערכת ההפעלה DOS. אופן פעולה זה מונע בעדם את היכולת לפנות למרחב זיכרון שמעבר ל-1MByte. בכדי לאפשר למעבד 80286 לפנות למרחב זיכרון מעל גבול זה, יש צורך לשנות את אופן פעולתו משיטת Real Mode לשיטת Protected Mode. פעילות של המעבד בשיטת מוגנת מאפשרת לו לפנות למרחב הזיכרון מעבר ל-1MByte הראשונים של מרחב הזיכרון. כמובן, שבסיום הפעילות של המעבד יש להחזיר את אופן הפעולה לשיטת Real Mode, על מנת שמערכת המחשב תוכל להמשיך לפעול תחת מערכת ההפעלה DOS בצורה תקינה.

תכניות המשתמש מיועדות לפעול בצורה תואמת לחלוטין בלא תלות בסוג המחשב שבו הן מתבצעות (XT, AT, PS/2). לשם כך נוצרו תקנים המאפשרים לתכניות השונות לפנות לזיכרון ההרחבה בצורה תקנית ומוסכמת מראש כדי שתתאפשר פעילות תקינה שלהן. סוגי זיכרונות ההרחבה השונים וצורת הפניה אליהם מפורטים להלן.

2.5.1 זיכרון הרחבה מסוג Expanded Memory

השם המקובל לזיכרון זה הוא EMS (Expanded Memory Specifications). התקן מגדיר שתפקידו לאפשר לתכניות גישה לזיכרון גדול מ-640KByte. באמצעותו ניתן להרחיב את הזיכרון הראשי של המחשב האישי באופן שכמעט ולא יפגע בביצועי המחשב. מהירות הביצוע של מערכת המחשב נשמרת, כי מהירות הגישה לזיכרון ההרחבה זהה למהירות הגישה לזיכרון הראשי עד ל-640KByte (קיימת תוספת השהיה קטנה של מספר פקודות מכונה על כל מעבר בין דפי זיכרון ההרחבה, כפי שיוסבר בהמשך). תקן זה שימש ליצרנים שונים בסיס לבניית כרטיסי הרחבה שונים הקיימים כיום בשוק.



ניהול זיכרון הרחבה – Expanded Memory

2.5.1.1 חומרה

זיכרון ההרחבה (ראה תרשים) יכול להיות בגודל של 64KByte ועד 32MByte. בכל עת מערכת ההפעלה והתכניות השונות יכולות "לראות" חלון בגודל 64KByte בלבד מתוך כלל הזיכרון הזה. על כן, כל גישה לקטע כלשהו של זיכרון ההרחבה אשר אינו מופיע בחלון, מחייבת פעולת מיתוג בחומרה כדי להציג את קטע הזיכרון המבוקש בחלון של זיכרון ההרחבה. פעולת המיתוג של הדף המבוקש לחלון נקראת Switched Memory Banks. היא מחייבת ביצוע של מספר פעולות מעבד.

זיכרון ההרחבה צורך לפחות 64KByte ממרחב הזיכרון של המעבד (1MByte ראשונים של המחשב). איזור זה של 64KByte מחולק ל-4 חלקים בגודל של 16KByte כל אחד, הנקראים **דפי זיכרון** (Memory pages). כל דף בחלון מקשר באופן פיסי בין מרחב הזיכרון של המחשב שאליו יכול המעבד לפנות לבין זיכרון ההרחבה. פניה זו אפשרית מכיון שמרחב כתובות זה נכלל בתוך 1MByte הראשונים של הזיכרון, אשר אליו יכולים לגשת כל המעבדים שאנו עוסקים בהם.

התקן המלא להרחבת זיכרון מסוג Expanded נקרא Lim 4.0. על פי הוראות תקן זה בונים יצרני חומרה את כרטיסי זיכרון ההרחבה ויצרני תוכנה - את היישומים המסוגלים לנצל זיכרון זה. התקן מאפשר לכרטיסי החומרה לנצל את כל איזורי הזיכרון הפנויים באיזור החומרה (מרחב זיכרון 640KB-1MB). התקן גם מאפשר להשתמש במרחב הזיכרון שבין 256KB ל-640KB כחלק מזיכרון ההרחבה. כאשר עושים שימוש זה מוציאים את רכיבי הזיכרון אשר ממופים במרחב כתובות 256KB-640KB מהלוח הראשי ואז הזיכרון שמעל 256KB קיים על גבי כרטיס ההרחבה.

מרבית כרטיסי זיכרון ההרחבה אינם משתמשים בכל האפשרויות שמגדיר תקן Lim 4.0. כרטיסי החומרה המשוכללים משתמשים ברוב זיכרון החומרה הפנוי (עד 128KByte) ובאיזור הזיכרון DRam בכתובות 512K-640K. על ידי שימוש זה ניתן ליישם חלון זיכרון הרחבה בגודל של 256KByte. כרטיסי הרחבה פשוטים יותר משתמשים בחלון של 64KByte בלבד מתוך מרחב החומרה. חלקם של כרטיסי חומרה אלה אינו תואם את תקן Lim 4.0 המורחב ולכן יש לבדוק כרטיסים אלה בצורה יסודית ולרכוש אותם רק לאחר שהם פועלים באופן תקין עם התכניות שבהן אנו מעוניינים להשתמש.

בתרשים ניתן תיאור סכימתי של מערכת הרחבת זיכרון ומיתוג דפים. על פי תרשים זה מופיעים בחלון מול מערכת ההפעלה הדפים הפיסיים הבאים:

דף מספר 0 בחלון הינו דף פיסי 4 בזיכרון ההרחבה.
דף מספר 1 בחלון הינו דף פיסי 8 בזיכרון ההרחבה.
דף מספר 2 בחלון הינו דף פיסי 9 בזיכרון ההרחבה.
דף מספר 3 בחלון הינו דף פיסי C בזיכרון ההרחבה (הספירה בשיטה הקסה-דצימלית)

המיתוג של הדפים אינו קבוע, אלא משתנה דינמית על פי צורכי התכניות ומערכת ההפעלה. תכנית ניהול הזיכרון (EMM) מנהלת עבור התכניות השונות את זיכרון ההרחבה ומספקת כלי תוכנה יעיל לצורך גישה לזיכרון זה.

זיכרון Expanded נפוץ ביותר בעבודה במערכות המחשבים ומרבית התכניות הקיימות בשוק ואלו שנכתבות, נבנות עם יכולת להשתמש בו. מעט תכניות יודעות להפעיל בצורה נכונה גם זיכרון Extended המוסבר בהמשך.

2.5.1.2 תוכנה

מרחב זיכרון ההרחבה מחולק **לדפים** (Pages) של 16KByte כל אחד, ללא תלות בגודלו. על מנת לאפשר מיתוג של דפים לחלון, להקצות דפי זיכרון פנויים ולפנות דפים שהתפנו, קיימים מעגל חומרה שמאפשר למתג כל אחד מהדפים לחלון ותכנית שמנהלת את הדפים. **התכנית לניהול הזיכרון** - EMM (Exapnded Memory Manager) מטפלת בניהול תפוסת זיכרון ההרחבה ובמיתוג הדפים הדרושים לחלון הפיסי. הדבר נעשה באמצעות פקודה מתאימה למעגל החומרה של הזיכרון.

תהליך העבודה של תכנית EMM מתבצע באופן הבא: התכנית מחלקת מספר לכל דף של זיכרון ההרחבה ויוצרת טבלת הקצאה של דפים (תפוס/פנוי). היא ממתגת 4 דפים מהזיכרון לחלון (שגודלו 64KByte) המופיע במרחב 1MByte של המעבד. הפעילות בחלון מתבצעת במהירות שבה מתבצעת הפעילות בזיכרון DRam הראשי.

כאשר תכנית יישום צריכה דף אשר נמצא בזיכרון ההרחבה ושאינו מופיע בחלון, היא מבקשת את הדף הזה מתכנית הניהול (EMM) באמצעות מנגנון הפסיקה (פסיקה 67Hex). תכנית ניהול זיכרון ההרחבה מקבלת את בקשת תכנית היישום, ממתגת את הדף המבוקש לאחד מהדפים בחלון וחוזרת לתכנית היישום. ברגע שתכנית המשתמש קיבלה אישור שהדף המבוקש נמצא בחלון, היא יכולה לקרוא/לכתוב לדף זה, בדיוק באותו האופן שבו היא פועלת מול הזיכרון הראשי. במרחב הזיכרון של המעבד מופיע בו זמנית מול מערכת ההפעלה ותכניות המשתמש חלון בגודל 64KByte מזיכרון ההרחבה, בלי תלות בגודלו הפיסי של הזיכרון. באופן זה מופיע מול מערכת ההפעלה והתכניות היישומיות המתבצעות במחשב, זיכרון RAM גדול בהרבה מהזיכרון הקיים בפועל במחשב.

עם זאת, למרות היתרונות הגדולים הגלומים בזיכרון ההרחבה, אין זה מספיק לחבר זיכרון כזה למחשב. על מנת שניתן יהיה להשתמש בו כראוי, צריכה התכנית לדעת כיצד להפעיל את מנגנון החלפת הדפים. תכנית אשר לא נבנתה עם יכולת שימוש בזיכרון ההרחבה, לא תוכל לנצל אותו! מערכת ההפעלה DOS (החל מגירסה 3.3) מאפשרת לתכניות שונות של מערכת ההפעלה שימוש חלקי בזיכרון ההרחבה. כך היא מקטינה את תפוסת הזיכרון הראשי במחשב. לדוגמה פקודת BUFFERS בגירסה 4.01, תכנית SMARTDRV שהיא תכנית מטמון דיסק (Disk Cache), תכנית RAMDRIVE (בגירסה 3.3 - תכנית VDISK). תוכנות יישומיות שונות אחרות, כמו Quattro PRO, Lotus 123 ואחרות יודעות לנצל את זיכרון ההרחבה.

גם במחשבי AT ומעלה ניתן להשתמש בזיכרון Expanded באותו אופן כמו במחשבי XT המבוססים על מעבדי 8088/8086.

2.5.2 זיכרון הרחבה מסוג Extended Memory

במחשבי AT המבוססים על מעבד 80286, או במערכות 386 המבוססות על מעבדי 386, ניתן להשתמש בזיכרון הרחבה מסוג **Expanded**. במערכות אלו גם פשוט יותר וכדאי לנצל את מרחב הזיכרון הגדול יותר מעל 1MByte שניתן לחבר למעבד באופן פיסי. למעבד 80286 אומנם ניתן לחבר עד 16MByte, אך מערכת ההפעלה DOS אינה מסוגלת לנצל מרחב זה ומפעילה את המעבד באופן פעולה Real Mode, המאפשר שימוש במרחב של 1MByte בלבד.

קיימות שתי אפשרויות לשימוש בזיכרון **Extended** המחובר פיסית במרחב הזיכרון שמעל 1MByte:

א. העברת המעבד תוך כדי עבודה, מאופן Real Mode וחזרה. העברה זאת נעשית על ידי התכנית המתבצעת במחשב. התהליך מורכב ומסובך ודורש מהתכנית שליטה בהפעלת המערכת ברמת מכונה. לכן, תכניות המשתמשות בזיכרון Extended באופן זה אינן נפוצות. מערכת בסיס נתונים Oracle, למשל, משתמשת בצורת עבודה זו.

ב. שימוש בתכניות שירות של ROM BIOS. תכניות שירות אלו מאפשרות להעביר בלוקים של זיכרון בגודל כלשהו בין מרחב הזיכרון עד 1MByte לבין מרחב זיכרון ההרחבה Extended Memory שמעל 1MByte. יתרון תכניות שירות של BIOS על פני זיכרון Expanded Memory הוא בכך שגודל בלוק הנתונים אינו מוגבל או קבוע, כמו דפי 16KByte בזיכרון Expanded (תכנית השירות של ROM BIOS המשמשת לצורכי העברת בלוקים של זיכרון הרחבה היא פסיקה 15Hex: פונקציה 87Hex - תוכן AH בקריאה לפסיקה). המנגנון פשוט להפעלה וגלוי לכל תכנית שפועלת במערכת ורוצה להשתמש ב-Extended Memory.

במחשבים המבוססים על מחשבי 80286 ודגמים מתקדמים יותר, ניתן לחבר על הלוח הראשי עד 16MByte זיכרון ראשי, ללא הוספה של כרטיסי הרחבה כלשהם, או הוספה של Drivers בתוכנה לשם טיפול בזיכרון זה.

מספר תכניות הפעלה להתקנים (Device Drivers) מאפשרות לדמות את זיכרון ההרחבה Extended לזיכרון ההרחבה Expanded. ההדמייה יעילה עבור תכניות ופקודות DOS, אשר אינן יודעות לנצל זיכרון הרחבה מסוג Extended, אך יודעות לנצל זיכרון הרחבה מסוג Expanded. דוגמאות לתכניות אלו: תכנית HIMEM.SYS של מערכת ההפעלה DOS, תכנית EMMA.SYS, תכנית QEXT המתלווה ל-DeskView ועוד. החיסרון העיקרי נובע מכך, שלא כל סוגי הלוחות הראשיים אכן פועלים בצורה אחידה תחת תכניות שירות אלה, ולכן לא תמיד ניתן לנצל אותו בצורה אמינה!

אופן הגישה לנתון בזיכרון Expanded שונה מאופן הגישה לזיכרון Extended. הבדלים אלה הם:

זיכרון Expanded	**זיכרון Extended**
1) מצא את הדף בזיכרון ההרחבה שבו נמצא הנתון המבוקש.	1) חשב כתובת של תא בזיכרון ההרחבה.
2) מתג את הדף על מנת שיופיע בחלון הזיכרון באמצעות קריאה לתכנית ניהול 15Hex.EMM	2) כתוב נתון לזיכרון בצורה ישירה (באמצעות מנגנון הפסיקה).
3) חשב כתובת של נתון בתוך הדף.	
4) רשום נתון לדף.	

שני סוגי הזיכרון האלה שונים זה מזה **במאפיינים נוספים:**

זיכרון Expanded	**זיכרון Extended**
1) זיכרון יקר יחסית, כאשר הוא מתחבר ככרטיס נוסף במערכת. מעגל החומרה מורכב יותר לצורך המיתוג.	1) זיכרון זול יחסית. מעגל חומרה פשוט יחסית.
2) אם מתחבר דרך מחברי ההרחבה הוא פועל במהירות BUS. לכן, במחשב מהיר הגישה לזיכרון זה איטית בהרבה מגישה לזיכרון רגיל המחובר למעבד הראשי. בחיבור על הלוח הראשי מתוסף זמן מיתוג הדף בלבד.	2) פועל במהירות גבוהה, מכיון שקשור ישירות למעבד ולכן לא מוריד מביצועי המחשב.
3) הזיכרון מחולק ליחידות של 16KByte. מתאים לעבודה בשיטת דפים (Paging) ב-80386 ומעלה.	3) אין הגבלת גודל. גודל גוש הזיכרון גמיש וניתן לשינוי.
4) המעבד "רואה" בכל רגע רק חלק מזיכרון ההרחבה, אשר גודלו נקבע לפי גודל החלון. במערכת הפועלת בשיטת Virtual Memory אין משמעות רבה לעבודה זו, כי בלאו הכי חלק מהתוכנה איננו קיים בזיכרון הראשי.	4) יכול לפעול כזיכרון רגיל הקשור למעבד. המעבד "רואה" כל הזמן את כל הזיכרון.
5) תואם לעבודה בכל המחשבים (צריך כמובן את כרטיס ההרחבה).	5) תואם למחשבי 286 ומעלה בלבד.
6) מרבית תכניות היישומים משתמשות בו, אם הוא קיים במערכת.	6) אין כמעט תכניות אשר משתמשות בו. לכן צריך תכנית תיאום להצגתו כזיכרון Expanded.

שני הסעיפים האחרונים הם המשמעותיים ביותר למשתמש.

2.5.3 זיכרון צללים - Shadow RAM

זיכרון הצללים פועל במחשבי 80286 ומעלה בלבד. כשמו, הוא כולל העתק של תכניות שירות לשם הפעלתן במהירות רבה יותר.

במספר מחשבים הקיימים בשוק רכיבי זיכרון ROM BIOS הינם איטיים, ואינם מסוגלים לפעול במהירות שבה פועל המעבד. פעולה איטית זאת מחייבת שימוש במנגנון מחזורי המתנה (Wait State) שמאט את פעולת מערכת המחשב. על כן, טעינה של תכניות השירות של ה-BIOS לזיכרון DRam (המהיר יותר), תשפר את ביצועי המערכת – בתנאי שה-DRam אכן פועל במהירות גבוהה יותר ממהירות זיכרון BIOS.

אם טעינת תכניות השירות נעשית תוך ניצול של מרחב הזיכרון הראשי עד 640KByte, שיפור מהירות הביצוע של המערכת יהיה על חשבון גודל הזיכרון הפנוי במחשב. ישנם לוחות שמקציבים זיכרון DRam נוסף במרחב של 1MByte. במקרה זה, אם נטען את תכניות השירות של BIOS ל-DRam מעבר ל-640KByte, נשפר את מהירות הביצוע ולא נגזול חלק ממרחב הזיכרון הראשי.

במחשבים מסוימים, שבהם מותקן זיכרון צללים, קיימת אפשרות לבטל את השימוש בו ובכך לחסוך במקום בזיכרון הראשי. במחשבי 386, שבהם אפשר לעשות זאת, טוענים את תוכנת הטיפול במסך ו/או את כל תכניות ROM BIOS על חשבון חלק מזיכרון ההרחבה, ולא על חשבון הזיכרון שמתחת ל-640KByte. שימוש ב-Shadow Ram אפשרי לכל תכניות ה-ROM BIOS או אך ורק לתכנית התצוגה המטפלות בהצגת המידע על גבי המסך. שימוש ב- Shadow Ram במערכת מחשב איטית מבוססת סביב מעבד 8088, לא יתרום להגברת ביצועי המערכת.

2.5.4 זיכרון מטמון – Cache Memory

שיטת זיכרון מטמון ישימה במחשבי 80386 ומעלה. זהו זיכרון מהיר ביותר, שזמן הגישה שלו 15-40nsec, לעומת כ-100nsec בזיכרון RAM רגיל. תפקידו להיות חיץ בין זיכרון DRam האיטי לבין המעבד המהיר, כדי לאפשר למעבד לפעול במהירות העבודה שלו ללא מחזורי המתנה.

זיכרון המטמון מחזיק נתונים מתוך זיכרון DRam בשיטת ניהול אשר דואגת לזמינות מירבית שלהם עבור המעבד. כאשר המעבד זקוק לפקודות תכנית או לנתונים הוא יגש תחילה לזיכרון המטמון. ברוב המקרים הוא ימצא אותם שם ולא יצטרך לפנות לזיכרון הראשי האיטי יותר, פעולה הכרוכה במחזורי המתנה המאיטים את התפוקה.

מן הראוי לציין ששיטת זיכרון המטמון מקובלת בכל המחשבים הגדולים.

מה מתבצע על ידי זיכרון המטמון?

אם המעבד צריך קטע תכנית או נתון כלשהו מהזיכרון הראשי (DRam) אשר לא נמצא בזיכרון המטמון, הוא יקרא אותו מהזיכרון הראשי במהירות הגישה של הזיכרון הראשי ויעביר אותו לזיכרון המטמון ולמעבד. במקרה זה אין כל

חיסכון בזמן גישה. אולם אם הנתון המבוקש על ידי המעבד נמצא בזיכרון המטמון יש חיסכון רב בזמן הגישה והנתון יעבור למעבד ללא השהיה.

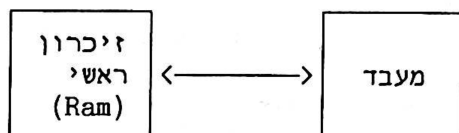
גודלו של זיכרון המטמון קטן יחסית לגודל הזיכרון הראשי (8-256Kbyte). אם כך, מתבקשת השאלה איך יודע המעבד אילו נתונים לשים בזיכרון המטמון בהנחה שיזדקק להם בטווח הקרוב, כי הרי לא ניתן לשים את כל תוכן הזיכרון הראשי בתוך זיכרון המטמון?!

ההחלטה של מנגנון ניהול זיכרון המטמון מסתמך על **עיקרון המקומיות** (Locality). עיקרון המקומיות בתוכנה קובע שבאחוז גבוה ביותר של מקרים, הנתון הבא שאותו יבקש המעבד יהיה הנתון הנמצא בכתובת העוקבת לנתון הנוכחי שאליו הוא ניגש, או בסביבה הקרובה של הנתון הנוכחי. לכן, אם הנתונים העוקבים לנתון הנוכחי יהיו בזיכרון המטמון, נגדיל את הסיכויים לכך שהמעבד ימצא אותם שם. הימצאות של הנתונים בזיכרון המטמון מקטינה את הצורך לגשת לזיכרון הראשי ותורמת לכך שהמערכת תבצע את התכניות במהירות גבוהה יותר מאשר ללא זיכרון מטמון.

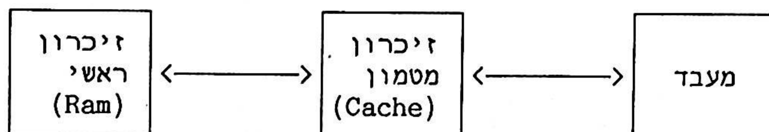
מימוש זיכרון מטמון במחשבים אישיים

במערכות מבוססות על מעבדי 80386 קיים בדרך כלל מעבד נוסף, Intel82385 המשמש כבקר להפעלת זיכרון המטמון (בקר 82395 במעבד 80386SX). בקר זה תומך בזיכרון מטמון בגודל של 32KByte. עבור כל נתון שומר זיכרון המטמון גם את ערך הנתון וגם את כתובת התא שבו הוא נמצא בזיכרון הראשי. ערכים אלה מאפשרים לעקוב אחר תפוסת זיכרון המטמון ותכולתו. זיכרון המטמון "שקוף" לתוכנה וכל הפעילות מולו נעשית על ידי המעבד ובקר זיכרון המטמון.

במערכות שמבוססות על מעבדי 80486 זיכרון המטמון, אשר כולל את הבקר וזיכרון בסיסי של 8KByte, הינו חלק מהמעבד. שילוב זה חוסך חיבורים חיצוניים ומאפשר למנגנון זה לפעול במהירות גבוהה לשיפור ביצועי המערכת. גודלי זיכרון מטמון המקובלים הינם 8KByte עד 256KByte.



מערכת מחשב ללא זיכרון מטמון



מערכת מחשב עם זיכרון מטמון

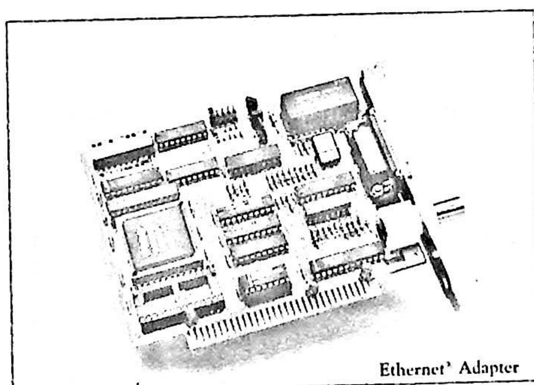
אין יחס ישיר בין גודל זיכרון המטמון לרמת ביצועיו. המדד לאיכות ביצועי זיכרון המטמון נקרא **אחוז הפגיעה** (Hit Ratio), אשר מבוסס על אלגוריתם חישובי. אחוז הפגיעה מודד כמה גישות לזיכרון המטמון נענו מתוכו לעומת מספר הפניות הכולל. אחוז הפגיעה המקובל בזיכרון מטמון של 8KByte, כפי שקיים ב-486, הוא 70-80%. הרחבה של זיכרון המטמון ל-32KByte תעלה את אחוז הפגיעה ל-85-90%. הרחבת זיכרון המטמון מעבר ל-32KByte לא תמיד תשפר את ביצועי המערכת, אלא במקרים שבהם יש לתכנית "התנהגות" כזו הפונה באקראי למקומות שונים בזיכרון ולא ברצף, כמקובל ברוב התכניות. לזיכרון מטמון גדול יש יתרון רב במערכות מרובות משימות (Multi tasking) לעומת מערכות הפועלות עם משימה אחת תחת מערכת ההפעלה DOS.

2.6 מרחב הקלט/פלט במערכת

בנוסף למרחב הזיכרון הבסיסי של 1MByte קיים במערכת המחשב מרחב זיכרון לקלט/פלט (I/O). למרחב זה מחוברים מרבית ההתקנים וכרטיסי התיאום שבמערכת. המעבד יכול לגשת בתוכנה למרחב קלט/פלט של 64KByte (סיביות A0-A15 של ערוץ הכתובות). במערכות הפועלות עם ISA BUS הוגבל מרחב זה ל-1024 כתובות בלבד במרחב 0000-03FFHex.

הגבלת מרחב הזיכרון נובעת מכך שכל הכרטיסים המחוברים למרחב הקלט/פלט מתייחסים אך ורק להדקים A0-A9 של ערוץ הכתובות (בעזרת 10 סיביות ניתן לגשת ל-1024 צירופים שונים בלבד). לפיכך, כתובת קלט/פלט 0000, או כתובת קלט/פלט 400Hex, או כתובת קלט/פלט 800Hex וכו' הן אותה כתובת. הכתובות זהות זו לזו מכיון שסיביות A0-A9 בכתובות אלו זהות, וכרטיס החומרה מתייחס אך ורק לסיביות אלו!

מרחב הקלט/פלט מאפשר חיבור התקנים ורכיבי תמיכה ועזר למעבד. חלקם של רכיבים/התקנים אלה נמצא על גבי הלוח הראשי וחלקם על גבי כרטיסי תיאום המתחברים למערכת באמצעות מחברי ההרחבה. בתהליך האתחול של המחשב, פונה המעבד להתקנים אלה אשר ממופים במרחב הקלט/פלט. תכנית האתחול "מחפשת" רכיבים/התקנים אלה על פי כתובתם הידועה מראש ומוגדרת בטבלה ומכינה אותם לעבודה בצורה הרצויה למערכת המחשב. הטבלה שלהלן מפרטת את רכיבי העזר והכרטיסים השונים הנמצאים במרחב הקלט/פלט:



תיאור הרכיב / כרטיס	מערכות מבוססות על מעבדי 8088/8086	מערכות AT מעבדי 80286 ומעלה
בקר DMA רכיב 8237 גישה ישירה	000-00F	000-01F
בקר פסיקות ראשון 8259-A	020-021	020-03F
זמנן (TIMER) 8253 ו-8254	040-043	040-05F
PPI רכיב 8255-A	060-063	לא קיים
מקלדת רכיב 8042	לא קיים	060-06F
שעון זמן אמיתי, מיסוך פסיקת NMI רכיב MC146818	רק NMI 0A0-0AF	070-07F
אוגר דפים לבקר DMA	080-083	080-09F
בקר פסיקות שני 8259-A	לא קיים	0A0-0BF
בקר 8237 DMA שני	לא קיים	0C0-0DF
ניקוי (CLEAR) למעבד מתמטי 80287	לא קיים	0F0
אתחול (RESET) למעבד מתמטי 80287	לא קיים	0F1
מעבד מתמטי 80287	לא קיים	0F8-0FF
דיסק קשיח FIXED DISK	320-32F	1F0-1F8
כרטיס מוט משחק (Game I/O)	200-20F	200-207
כרטיס מדפסת מקבילית שני	278-27F	278-27F
כרטיס טורי משני	2F8-2FF	2F8-2FF
כרטיס אב-טיפוס לפיתוח	300-31F	300-31F
כרטיס מדפסת מקבילית ראשון	378-37F	378-37F
כרטיס תקשורת סינכרוני, SDLC שני	380-38F	380-38F
כרטיס תקשורת סינכרוני ראשון	3A0-3AF	3A0-3AF
כרטיס מתאם מסך מונוכרום	3B0-3BF	3B0-3BF
כרטיס מתאם מסך צבעוני	3D0-3DF	3D0-3DF
בקר כונני דיסקטים	3F0-3F7	3F0-3F7
כרטיס טורי ראשון (אסינכרוני)	3F8-3FF	3F8-3FF

* כל הכתובות בטבלה הינן בהקסה

מיפוי כתובות מרחב קלט/פלט (I/O)

2.7 רכיבים מתוכנתים על הלוח הראשי

על הלוח הראשי קיימים מספר רכיבי חומרה הניתנים לשליטה בתוכנה. הם מחוברים למעבד ויחד עימו יוצרים את המחשב האישי.

במחשבים תואמים חדישים לא נמצא את מספרי הרכיבים המפורטים בהמשך. עם ההתפתחות המואצת בשטח המחשבים, קיימת כיום אפשרות לדחוס יותר רכיבים לתוך שטח קטן ולכן נראה רכיבים המשלבים בתוכם מספר רב של רכיבים מבין אלה המפורטים בטבלה למעלה או בהמשך. לדוגמה, רכיב VL82C100 ב-AT מכיל 2 בקרי DMA 8237, 2 בקרי פסיקות 8259, רכיב Timer 8254 ומעגלים לוגיים נוספים. על כן, ההתייחסות היא למספרים ולשמות על פי תפקודם של

הרכיבים במערכת המחשב, ולא על פי האריזה שבתוכה הם כלולים. תפקוד הרכיבים, כתובתם במרחב הקלט/פלט והתכנות שלהם, נשארים ללא שינוי, גם אם צורת האריזה משתנה.

את אופן התכנות של כל רכיב והאפיון של פקודות התכנות ניתן למצוא בספר Intel Microcomputer Handbook. את פקודות האתחול/תכנות של הרכיב יש לשלוח לכתובת המופיעה בטבלה לעיל.

2.7.1 זמנן - TIMER (רכיב 8253 או 8254)

רכיב זה הינו בר תכנות. הוא מכיל שלושה מונים/זמננים Timer/Counters המשמשים את המערכת לצרכים שונים. בשלב האתחול, אחראית תוכנת ROM BIOS לשלוח לרכיב זה פקודות על מנת להפעילו בצורה הרצויה למחשב.

המונים/זמננים משמשים את המחשב האישי לצרכים הבאים:

מונה 0: משמש כמונה זמן כללי וכקוצב זמן בסיסי. יציאת המונה מחוברת לבקר הפסיקות במערכת (פסיקה 8). כל זמן קצוב (18.2 פעם בשנייה) גורמת יציאה זאת להפסקה בפעילות המעבד לשם עדכון "שעון זמן" (שעות, דקות, שניות ותאריך). כשיש פסיקת זמן קורא המחשב לפסיקה 1CHex. המתכנת יכול להחליף את ווקטור הפסיקה 1CHex בכתובת של תכנית אחרת רצויה. כך, בכל פסיקת זמן יבצע המחשב קריאה גם לתכנית המשתמש, כמו לדוגמה תכנית המציגה את השעה והתאריך באופן קבוע על גבי המסך. ראה תכנית דוגמה בנספח.
הערה: מונה זה יוצר פסיקת חומרה. הסבר נוסף לפסיקות חומרה ראה בפרק המתאים.

הדקי מונה 0:

בקרת פעולה (GATE)	-	אין, פועל באופן קבוע.
תדר כניסה	-	1.19 MHz
מוצא	-	מחובר להדק בקשת פסיקה 0 בבקר פסיקות.

מונה 1: משמש למערכת כבקשת שירות DMA מערוץ רביעי של בקר DMA (ראה בסעיף הבא). ערוץ DMA משמש את מערכת המחשב לרענון זיכרון DRam. בכל פרק זמן קבוע מבקש המונה בקשת DMA ובכך הוא מפעיל את מנגנון הרענון של הזיכרון כדי למנוע אובדן תוכן. הפסקת פעולת המונה תגרום להפסקת הפעולה של המחשב ותחייב פעולת אתחול של המערכת.

הדקי מונה 1:

בקרת פעולה (GATE)	-	אין, פועל באופן קבוע.
תדר כניסה	-	1.19MHz
מוצא	-	מחובר להדק בקשת רענון זיכרון דינמי.

מונה 2: משמש את מערכת המחשב האישי ליצירת טונים וקולות המושמעים ברמקול המחשב. מוצא המונה מחובר לרמקול של המחשב דרך מערכת בקרה להשתקת הרמקול באמצעות תוכנה, ועל כן תדר המופק באמצעות מונה זה משמיע צלילים ברמקול. המונה משמש גם כקוצב זמן לצורך חישובי זמנים והשהיות במערכת, אך במקרה זה הדבר מחייב השתקת הרמקול.

הדקי מונה 2:

בקרת פעולה (GATE) - בקרה על ידי סיביות 0 של כניסה 61Hex
תדר כניסה - 1.19 MHz
מוצא - רמקול המחשב האישי (דרך רכיב הגברה) ורכיב בקרה לצורך חיבור או ניתוק של הרמקול (סיבית 1 של מרחב קלט/פלט 61Hex)

2.7.2 בקר גישה ישירה לזיכרון - DMA (רכיב 8237)

פעולות העברה של נתונים בין הזיכרון הראשי של המחשב (RAM) לבין מערכות הדיסקים/דיסקטים במערכת אינן מחייבות התערבות של המעבד הראשי. על מנת לבצע פעולות העברה אלו, על המעבד לבצע תכנית שלמה שנמשכת זמן רב. בכדי לחסוך התערבות המעבד בהעברת הנתונים מהזיכרון ליחידות המחוברות במרחב הקלט/פלט ולהיפך, משתמש המעבד בבקר DMA (Direct Memory Access), אשר מאפשר להעביר נתונים מהזיכרון ואליו ללא התערבות המעבד.

כדי להעביר נתונים המעבד מוסר לרכיב DMA את ההוראות לביצוע ההעברה:

- כמה בתים יש להעביר מהזיכרון, או אל הזיכרון,
- הכתובת ההתחלתית של הבלוק שיש להעביר,
- צורת העבודה שבה יש לנקוט לצורך ההעברה (העברת בית והחזרת שליטה למעבד, או העברה רצופה של הבלוק ללא התערבות המעבד).

בגמר שלב זה, בקר DMA מבצע את ההעברה בעצמו, ללא התערבות המעבד, ובזמן קצר בהרבה מהזמן שהיה נדרש לבצע העברה על ידי המעבד. לצורך ההעברה בקר DMA חייב להשתמש בערוצי הכתובות והנתונים שדרכם מתקשר המעבד לעולם החיצוני. על כן, כאשר בקר DMA מעביר נתונים המעבד אינו יכול לבצע פעולות בערוצים השונים. על מנת להבדיל בין פעולת DMA מפעילות רגילה של המעבד, קיימת סיבית בקרה מיוחדת בערוץ הבקרה המציינת מתי בקר DMA מבצע פעילויות במערכת המחשב. כל מחזור DMA נמשך 5 מחזורי שעון לצורך העברה אחת של נתונים.

הערוץ הרביעי של בקר DMA משמש לרענון זיכרון DRam. הוא משתמש ב-4 מחזורי שעון לצורך מחזור רענון יחיד. במחשבי AT המבוססים על מעבדי 80286 ומעלה קיימים שני בקרי DMA.

ערוצי DMA ותפקידיהם:

ערוץ 0 - חופשי
ערוץ 1 - תקשורת SDLC כרטיס תקשורת סינכרוני הפועל בפרוטוקול SDLC (Synchronous Data Link Control)
ערוץ 2 - בקר דיסקים / דיסקטים.
ערוץ 3 - משמש לרענון של הזיכרון הדינמי במחשב. המונה 8253 מתוכנת לבקש בכל פרק זמן קבוע בקשת DMA מדומה מהבקר. הבקר פונה לזיכרון הדינמי הראשי בכל פרק זמן קבוע ומרענן את תוכנו כדי שלא יאבד.

במחשבי AT קיימים 4 ערוצי DMA נוספים. כל אחד מהם יכול לשמש להעברת נתונים ברוחב של 16 סיביות בין הזיכרון הראשי למרחב קלט/פלט ולהיפך. העברה של 16 סיביות הינה מהירה יותר מהעברה של 8 סיביות, כי בכל מחזור מועברים 2 בתים לעומת אחד.

ערוץ 4 - משמש לצורכי קישור לבקר DMA הראשון
ערוץ 5 - חופשי
ערוץ 6 - חופשי
ערוץ 7 - חופשי

2.7.3 בקר פסיקות - Interrupt Controller (רכיב 8259A)

אל בקר הפסיקות ניתן לחבר עד 8 התקנים (devices) שונים. באמצעות בקר הפסיקות יכולים התקנים אלה להפסיק את פעולת המעבד ולגרום לביצוע תכנית שירות אשר מטפלת בהתקן המבקש את הפסיקה. הפסיקה הינה הפרעה של הבקר למעבד במהלך פעילותו כדי להודיע לו, כי אחד ההתקנים הקשורים אליו מבקש שירות. למעבד יש אפשרות לשלוט בבקשות שירות של כל התקן וגם לחסום/לאפשר (enable/disable) בקשות אלו על פי הצורך. שליטת המעבד בבקר הפסיקות מושגת באמצעות כתיבת פקודה מתאימה לבקר הפסיקות.

הביצוע

תהליך בקשת הפסיקה של ההתקן:

1. ההתקן מפעיל את הדק REQ INTR שלו בבקר,
2. הבקר בודק אם להתקן מותר להפעיל פסיקה,
3. אם כן - הבקר מפעיל את הדק INTR במעבד,
4. לאחר שהמעבד מתפנה הוא נותן לבקר אישור לפסיקה,
5. הבקר שולח למעבד את כתובת הפסיקה הדרושה,
6. המעבד מבצע פסיקה.

פירוט הביצוע

כאשר אחד ההתקנים פונה לבקר הפסיקות ומבקש פסיקה ומוגדר כי מותר לו לבקש זאת, בקר הפסיקות מציג למעבד בקשת פסיקה (הדק INTR של המעבד). אם מנגנון הפסיקות של המעבד מאפשר לו לקבל בקשת פסיקה, המעבד ישלים את הפעולה הנוכחית שהוא מבצע ויאשר לבקר הפסיקות להודיע לו איזה התקן מבקש פסיקה. המעבד שולח את האישור לבקר הפסיקות באמצעות הדק INTR של המעבד ומקבל ממנו מספר המציין איזו תכנית שירות לבצע.

לפני שהמעבד ניגש לבצע את תכנית השירות לטיפול בפסיקה, הוא שומר את כתובת ההוראה הבאה שעליו לבצע בגמר הטיפול בבקשת השירות ואת אוגר הדגלים של המעבד. נתונים אלה נשמרים על מנת שהמעבד יוכל לחזור לנקודה שבה הפסיק את הפעילות הרגילה לאחר סיום השירות של הפסיקה. המעבד ממשיך לבצע את התכנית הקודמת מאותה נקודה מבלי שהמשתמש, או התכנית ירגישו בהפסקת הפעילות השוטפת ובטיפול בפסיקה.

כאשר בקר הפסיקות מקבל אישור פסיקה מהמעבד, הוא שולח למעבד מספר בן 8 סיביות המציין את מספר הפסיקה שיש לבצע. מספר בן 8 סיביות מאפשר

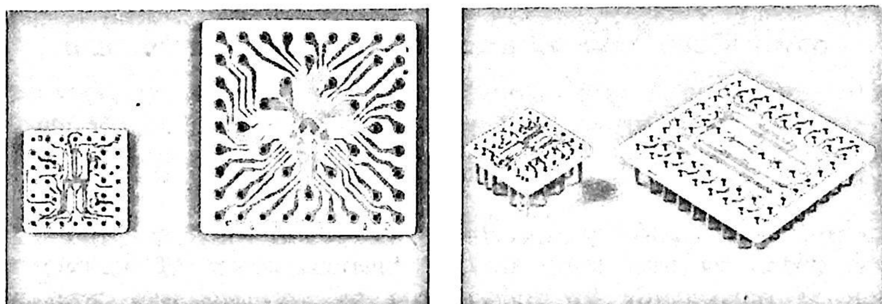
להגדיר מספרי פסיקות בין 00-FFHex, המתאימים בדיוק לגודל טבלת וקטורי הפסיקות. ואומנם, המעבד משתמש במספר הפסיקה כדי לפנות לכניסה המתאימה בטבלת וקטורי הפסיקות. כל כניסה מכילה הפניה לתכנית שירות אשר תטפל בפסיקה באמצעות כתובת התחלתית של הפסיקה במרחב הזיכרון של המעבד (CS:IP - כתובת אוגר מקטע ומצביע הוראות).

לביצוע תכנית הפסיקה המעבד פונה לכתובת המוגדרת על ידי אוגר המקטע ומצביע ההוראות (CS:IP) וממשיך לבצע את פעולתו מכתובת זאת. כל תכנית פסיקה חייבת לשמור את תוכן האוגרים שבהם היא משתמשת. לפני שהיא מחזירה את המעבד לפעילות הקודמת שלו, היא משחזרת את ערכם כפי שהיה לפני שהופעלה תכנית הטיפול בפסיקה. השמירה והשחזור מאפשרים להמשיך בפעילות העיבוד השוטפת מן המקום שהופסקה. תכנית הפסיקה חייבת להודיע לבקר הפסיקות כי הסתיים תהליך הטיפול בהתקן שביקש את הפסיקה, כדי לאפשר לו לחזור ולהמשיך בפעילות שהפסיק.

החזרת המעבד לפעילות שהפסיק נעשית באמצעות הפקודה IRET (חזור מפסיקה) אשר שייכת לאוסף הפקודות של המעבד. למעבד יש אפשרות לכתוב פקודת בקרה לבקר הפסיקות על מנת לחסום, או לאפשר בקשות פסיקה מהתקנים שונים על פי הצורך. יש לזכור, כי אם החסימה תיעשה בצורה בלתי מבוקרת ומבלי שיישמר המצב הקודם, סביר להניח כי תפגע הפעילות השוטפת של מערכת המחשב.

עבור כל התקן המקושר לבקר הפסיקות (התקנים 0-7) מוקצית סיבית בקרה כדי לאפשר, או לחסום, את בקשת הפסיקה של ההתקן. במחשבי AT ומעלה קיים בקר פסיקות 8259A נוסף, המאפשר לחבר 8 התקנים נוספים, או 16 התקנים לשני הבקרים. צורת הטיפול בבקר הפסיקות השני זהה לצורת הטיפול בבקר הראשון. ההבדל הוא בכך ששני בקרי הפסיקות ממופים בכתובות קלט/פלט שונות זו מזו. כניסות ההתקנים השונים בבקר הפסיקות מסודרות על פי דרגות עדיפות לבקשות פסיקה. העדיפות הגבוהה ביותר היא להתקן המחובר לכניסה 0 בבקר הפסיקות הראשון והנמוכה ביותר - להתקן המחובר לכניסה מספר 7 בבקר הפסיקות השני. התכנית המטפלת בפסיקת חומרה חייבת להודיע על סיום הטיפול בפסיקה. היא מעבירה מלת בקרה לבקר הפסיקות הראשון אם היא קבלה ממנו את הפסיקה. אם היתה פסיקה מבקר הפסיקות השני, היא חייבת להודיע על סיום הטיפול בפסיקה **לשני** בקרי הפסיקות, ולא רק לבקר השני.

פירוט של פסיקות חומרה ותוכנה במערכת המחשב ניתן לקבל באמצעות תכניות שירות שונות כמו Sysinfo של Norton גירסה 5, או PcTools.



מספר פסיקה בבקר	מספר פסיקה במחשב	תיאור הפסיקה
0	8	פסיקת שעון - מוצא TIMER 0
1	9	לוח מקשים
2	A	לוח הרחבה 4
3	B	תקשורת אסינכרונית 2 או
		תקשורת סינכרונית SDLC
4	C	תקשורת אסינכרונית 1 או
		תקשורת סינכרונית SDLC
5	D	בקר דיסק קשיח
6	E	בקר כונן דיסקטים
7	F	מדפסת מקבילית

מיפוי פסיקות במחשבי PC המבוססים על מעבדי 8088/8086

	מספר פסיקה בבקר	מספר פסיקה במחשב	תיאור הפסיקה
	0	8	פסיקת שעון מוצא TIMER 0
	1	9	מקלדת
בקר	2	A	פסיקה מבקר פסיקות שני
פסיקות	3	B	תקשורת אסינכרונית 2 או
ראשון			תקשורת סינכרונית SDLC
	4	C	תקשורת אסינכרונית 1 או
			תקשורת סינכרונית SDLC
	5	D	מדפסת מקבילית שניה
	6	E	בקר כונני דיסקטים
	7	F	מדפסת מקבילית ראשונה
	0	70 Hex	שעון זמן אמיתי
	1	71	מקושר בתוכנה לפסיקה A
בקר	2	72	שמור
פסיקות	3	73	שמור
שני	4	74	שמור
	5	75	מעבד מתמטי
	6	76	בקר דיסק קשיח
	7	77	שמור

מיפוי פסיקות במחשבי AT המבוססים על מעבד 80286 ומעלה

2.7.4 שעון זמן אמת - Real Time Clock

הרכיב "שעון זמן אמת" מאפשר לדעת את התאריך והשעה. הוא קיים כהתקן בסיסי במחשבי AT ומעלה ובמחשבי XT ניתן לקבל אותו על כרטיס הרחבה. הרכיב מכיל שעון זמן אמת עם גיבוי סוללה של 6V המעניקה לו אפשרות

לפעול גם כשמכבים את המתח למחשב. בלוחות של מחשב אישי המבוססים על מעבד 8088/8086 לא קיים שעון כזה על הלוח הראשי ועל כן צריך לחזור ולכוון את השעה והתאריך בכל אתחול של המחשב, אלא אם הוספנו כרטיס שעון זמן אמיתי חיצוני. רכיב זה מכיל בנוסף לשעון זמן אמת גם את ההתקנים הבאים:

* שעון אזעקה בר-תכנות המאפשר לבצע תכנית, או לקבל חיווי אזעקה לאחר פרק זמן שנתבקש מראש (עד 24 שעות).

* זיכרון RAM של 64 בתים, אשר ב-16 בתים מתוכם השעון שומר את הזמן והתאריך. המעבד משתמש ביתרת זיכרון זה על מנת לשמור את תצורת החומרה במחשב: סוגי כונני הדיסקטים המותקנים במערכת, סוג הדיסק הקשיח ועוד - ראה טבלה להלן.
  תצורת מערכת החומרה חיונית במערכות AT ומעלה שאין בהם מפסקים המגדירים את החומרה כמו ב-XT. כאשר הסוללה אינה תקינה מקבלים חיווי שגיאה במהלך תהליך האתחול, המתריע על טעות בהגדרת תצורת החומרה. טעות זאת נובעת מכך שזיכרון השעון נמחק כאשר מתח הסוללה לא הגיע אליו. פתרון הבעיה - על ידי החלפת הסוללה וכיוון מחדש של תצורת המחשב.

את תצורת המערכת הרשומה בזיכרון השעון ניתן לשנות בתהליך Setup של המחשב (מחשבי AT ומעלה) וניתן להציג אותה על גבי מסך המחשב בעזרת תכניות שירות שונות (כמו Norton). במחשבי AT ישנים של יבמ ניתן היה לבצע Setup אך ורק בעזרת תכנית דיאגנוסטיקה של יבמ. מחשבים חדישים מספקים אפשרות זאת בצורה נוחה וידידותית בתהליך האתחול על ידי תכנית ROM BIOS.

אפשר לגשת לשעון זמן אמת באמצעות מנגנון פסיקת שירות מספר 1AHex של ROM BIOS. אפשר גם לכתוב ישירות לזיכרון הרכיב, אשר ממופה בכתובות 70-71Hex במרחב הקלט/פלט. כתיבה/קריאה ישירה ממנו נעשים בשני שלבים:

* כתיבת הכתובת של התא הנדרש ב-RAM המגובה של השעון, אל בקר הזיכרון בכתובת קלט/פלט 70Hex.
* קריאה/כתיבה של הערך הנדרש מכתובת קלט/פלט 71Hex.

שינוי של מרבית הערכים בשעון זמן אמת לא ישפיעו מיידית על המחשב, אלא רק לאחר שנאתחל שוב את המחשב. מרבית הערכים נקראים על ידי המחשב רק פעם אחת בתהליך האתחול ואינם משמשים בעבודה שוטפת. לדוגמה, עדכון זמן בשעון זמן אמת על ידי כתיבת השעה לרכיב, ישפיע רק בפעם הבאה שנאתחל את המערכת ולכן, אם נקיש פקודת TIME במערכת ההפעלה DOS מיד לאחר ששינינו את השעה, לא נראה שינוי זה בזמן המערכת.

שירות פסיקת ROM BIOS לטיפול בשעון מבדילה בין עדכון/קריאת שעון מערכת לבין עדכון/קריאת שעון זמן אמת. תכנית ROM BIOS מספקת אמצעי גישה לשעון זמן אמת באמצעות מנגנון פסיקת שירות מספר 1AHex, אשר מאפשרת גם הפעלת שעון אזעקה (Alarm Clock). תכנית לדוגמה לטיפול בשעון זמן אמת ניתן למצוא בנספח תכניות דוגמה.

כתובת Hex	ייעוד בזיכרון RAM של שעון זמן אמת
* 00	שניות שעון
* 01	שניות שעון אזעקה
* 02	דקות שעון
* 03	דקות שעון אזעקה
* 04	שעות שעון
* 05	שעות שעון אזעקה
* 06	יום בשבוע
* 07	יום בחודש
* 08	חודש
* 09	שנה
* 0A	אוגר מצב A
* 0B	אוגר מצב B
* 0C	אוגר מצב C
* 0D	אוגר מצב D
* 0E	מצב דיאגנוסטיקה (בדיקה) של רכיב שעון זמן אמת
* 0F	מצב כבוי (Shutdown)
10	סוג כונן דיסקטים A ו-B
11	שמור
12	סוג דיסק קשיח ראשון ושני
13	שמור
14	ציוד מותקן (מספר כוננים, סוג מסך לצורך BOOT, מעבד מתמטי ועוד.
15	בית תחתון של גודל זיכרון בסיסי (עד 640KByte)
16	בית עליון של גודל זיכרון בסיסי
17	בית תחתון של גודל זיכרון הרחבה Extended
18	בית עליון של גודל זיכרון הרחבה Extended
19-2D	שמור
2E-2F	בקרת שגיאות בתוכן RAM של השעון (Checksum)
* 30	בית תחתון של גודל זיכרון הרחבה Expanded
* 31	בית עליון של גודל זיכרון הרחבה Expanded
* 32	ערך של המאה הנוכחית (המאה ה-20 במקרה זה)
* 33	דגלי מידע (מעודכנים בזמן BOOT)
34-3F	שמור

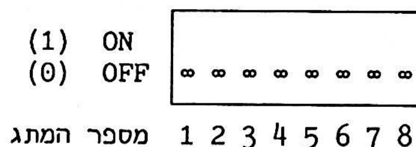
* ערך זה לא נכלל בחישוב בקרת שגיאות (Checksum)

מיפוי זיכרון של שעון זמן אמת

2.7.5 מתגים על הלוח הראשי

בחלק של הלוחות הראשיים המבוססים על מעבדים 8086/8088 קיימים 8 מתגים (Dip Switches) הקובעים את תצורת החומרה במערכת. בכל המערכות האלו תפקיד המתגים ומיקומם זהה כמעט לחלוטין. על מנת שמערכת המחשב תפעל בצורה תקינה יש לכוון מתגים אלה לתצורת החומרה הקיימת.

המתגים נמצאים באריזה אחת הכוללת 8 מתגים על פי הפירוט הבא:



מבנה רכיב מתגים על גבי לוח ראשי XT

מתג 1 - מצב נורמלי במצב OFF. בחלק מהמחשבים משמש לביטול מעגל בדיקת Parity. כאשר המתג במצב OFF, המחשב משתמש במעגל Parity לבדיקת תקינות זיכרון DRam. במקרה זה יש לשים רכיבי זיכרון גם ברכיב הזיכרון התשיעי המיועד לבדיקת Parity. אם רוצים לבטל בדיקת Parity יש להעביר את המתג למצב ON ואז אין חובה לשים רכיב זיכרון DRam תשיעי בכל שורה.

מתג 2 - משמש להגדרת מעבד מתמטי במערכת (8087).
OFF - קיים מעבד מתמטי במערכת.
ON - לא קיים מעבד מתמטי במערכת.
אם נגדיר כי על גבי הלוח הראשי קיים מעבד מתמטי למרות שבפועל הוא איננו קיים, יווצר מצב שגיאה ברוב המערכות והן לא תפעלנה. לעומת זאת, רוב התוכנות שיודעות להשתמש במעבד המתמטי, בודקות בעצמן אם הרכיב נמצא על גבי הלוח הראשי ולא מתייחסות למצב המתג.

מתג 3-4 - מתגים אלה משמשים לציון גודל זיכרון RAM במערכת. כדי להגדיר מצב מתגים אלה יש להסתכל בספר המחשב, כי ההגדרה הזו אינה תקנית ושונה בין לוחות ראשיים שונים. במרבית הלוחות החדשים אין משמעות למתגים אלה.

מתגים 5-6 - מגדירים את סוג מתאם המסך המחובר למחשב. אם מחובר יותר ממתאם מסך יחיד, הגדרה זו מציינת למחשב דרך איזה מתאם מסך עליו להציג את המידע – מסך ברירת המחדל של המערכת. כפי שראינו אפשר לחבר יותר מסוג מסך אחד מכיון שנתוני המסכים ממופים בכתובות שונות. אם נגדיר באמצעות המתגים סוג לא נכון של מתאם מסך, לא נראה דבר על גבי המסך.

המתג	5	6	
	OFF	OFF	- מתאם מסך מונוכרום (MGA) מחובר למערכת.
	ON	OFF	- מתאם מסך צבעוני (CGA). צורת הצגת המידע על גבי המסך: 40 תווים בשורה ב-25 שורות.
	OFF	ON	- מתאם מסך צבעוני (CGA). צורת ההצגה על גבי המסך: 80 תווים בשורה ב-25 שורות.
	ON	ON	- בלוחות ישנים המשמעות היא, שלמחשב לא מחובר מתאם מסך כלשהו. בלוחות חדשים יותר המתגים מציינים שמחובר מתאם מסך EGA.

מתגים 7-8 - קובעים את מספר כונני הדיסקטים (Floppy Disk) שמחוברים למחשב. כיוון של מתגים אלה בצורה לא נכונה יפגע בפעולה תקינה של כונני הדיסקטים. כיוון המתגים למספר כוננים קטן מכפי שקיים בפועל ימנע הפעלת כוננים מסוימים וכיוון למספר גבוה יותר עלול לגרום להצגת הודעת שגיאה על גבי המסך.

המתג	7	8	
	OFF	OFF	4 – כונני דיסקטים מחוברים למערכת.
	ON	OFF	3 – כונני דיסקטים מחוברים למערכת.
	OFF	ON	2 – כונני דיסקטים מחוברים למערכת.
	ON	ON	1 – כונן דיסקטים מחובר למערכת.

במקרה האחרון שבו מוגדר כונן אחד בלבד - פניה לכונן B, גם אם קיים כונן כזה, תגרור פניה לכונן A. אם לא קיים כונן B, הכיוון של המתגים האלה מונע טעויות כאשר פונים לכונן B, בכך שכל פניה ל-B נענית מכונן A. כיוון המתגים יעיל כאשר במערכת כונן יחיד ורוצים להעתיק קובץ יחיד מדיסקט לדיסקט. פקודת מערכת ההפעלה "COPY a:file-name b:" תאפשר העתקה בין דיסקטים.

בלוחות ראשיים של מחשבי AT מופיעה תצורת החומרה על ה-RAM המגובה של שעון זמן האמת. למרות זאת, נראה מגשרים שונים (Jumpers) על הלוח הראשי, הן במחשבי AT והן במחשבי PC המבוססים על מעבדי 8088/8086. תפקידם - לאפשר שליטה בתצורת החומרה ובאפיוני החומרה של המערכת. שימושים אופייניים למגשרים אלה:

* שליטה על התדר שבו המחשב יתחיל לפעול, במידה וקיימת אפשרות להפעיל את המחשב במספר מהירויות.

* שליטה על מספר מחזורי ההמתנה של המערכת (Wait States).

* תצורת זיכרון RAM על גבי הלוח הראשי, כמו לדוגמה: האם על גבי הלוח אפשר לשים 256KByte, 640KByte, או 1MByte. גישור זה קובע כיצד יש להתייחס לרכיבי הזיכרון שעל גבי הלוח ואילו רכיבים יש לשים לצורך פעולה תקינה.

* סוג רכיבי ROM BIOS המחוברים על גבי הלוח הראשי (קיבולת, מהירות וכו').

* סוג מתאם המסך המחובר למערכת.

* זמני גישה של רכיבי זיכרון DRam.

* אפשור / חסימה של זיכרון מסוג Shadow Ram.

אלה הם שימושים אופייניים בלבד, ואין זה מחייב שהמגשרים שהוזכרו יימצאו כולם או מקצתם בכל לוח ראשי. מידע מפורט על הלוח הראשי של כל מחשב ניתן למצוא בספר אשר מתאר את הלוח הראשי שלו.

2.7.6 מחברי הרחבה - Expansion Slots

על הלוח הראשי במחשבים המבוססים על מעבדי 8088/8086 קיימים מספר מחברי הרחבה (5 במחשבי IBM-PCG ו-8 - בתואמים ובמחשבי IBM-PC/XT). כל מחברי ההרחבה זהים זה לזה ונועדו לאפשר הרחבה של המערכת ולהוסיף מתאמים ובקרים ללוח הראשי לשם הפעלה תקינה ושוטפת שלה: בקר כונני דיסקטים, בקר מסך, בקר דיסק קשיח ועוד. התקשורת בין המעבד למתאמים והבקרים השונים נעשית באמצעות המחברים, אשר דרכם מתקבל גם המתח הדרוש להפעלתם.

המעבד מעביר דרך מחברי ההרחבה את פקודות ההפעלה למתאמים ועל ידי כך הוא מאפשר את העברת הנתונים בינו לבין כרטיסי התיאום ולהיפך. כל מחברי ההרחבה מאפשרים העברת מידע על גבי פס נתונים ברוחב של 8 סיביות (בית אחד). במערכות גדולות שיש לחבר אותן ליותר מ-8 התקנים חיצוניים, דרושים יותר מ-8 כרטיסי הרחבה. לכן קיימים בשוק מארזים המכילים ספק כוח וכרטיס תיאום שאפשר לחבר אותם למעבד במארז הראשי וכך להוסיף 12 מחברי הרחבה ולהגיע למספר כולל של 20 מחברים.

במחשבי AT ומחשבי 80386SX קיימים 8 מחברים משני סוגים:

- מחברי הרחבה רגילים לצורך חיבור מתאמים המעבירים נתונים ברוחב 8 סיביות, כמו במחשבים המבוססים על מעבד 8088.
- מחברים נוספים גדולים יותר, המאפשרים חיבור מתאמים המעבירים נתונים ברוחב של 16 סיביות (Word).

תוספת זו מאפשרת לנצל את תכונות מעבדי 80286 ו-80386SX ובכך לשפר את מהירות הביצוע של הפעולות במערכת על ידי העברה של 16 סיביות. המתאם, או הבקר, חייב לתמוך בסוג העברה כזה, גם אם לא כל המתאמים השונים יודעים לעשות זאת. מחברי ההרחבה עד 16 סיביות מאפשרים להתקשר למערכת בארכיטקטורת ערוצים הנקראת ISA BUS (Industry Standart Architecture) אשר מאפשרת לכרטיסי התיאום להתחבר למערכת המחשב ולעבוד במהירויות מירביות של 8-10MHz.

המהירות הנמוכה יחסית של הערוץ גורמת לקושי במערכות מחשבים מהירות, שבהן המעבד פועל ומתקשר עם הזיכרון הקיים על גבי הלוח הראשי במהירויות גבוהות (20-33 MHz). במערכות אלו, כל גישה של המעבד לעולם החיצוני דרך מחברי ההרחבה מחייבת שימוש במנגנוני המתנה, אשר גורמים להאטה בפעולת מערכת המחשב. המגבלה במהירות הערוץ נובעת מהמבנה הפנימי של מחברי ההרחבה, אשר אינם יכולים להעביר נתונים בקצבים שמעל 8-10 MHz.

במחשבים המבוססים על מעבד 80386DX נמצא מחברי הרחבה של 32 סיביות להעברת נתונים ברוחב זה בגישה אחת של המעבד להתקן. בנוסף למחברי 32 סיביות נמצא בהם גם מחברי 8 סיביות ומחברי 16 סיביות שנועדו לצורכי תאימות חומרה כלפי מטה. הארכיטקטורה המגדירה תקשורת ברוחב של 32 סיביות בין המעבד לכרטיסי התיאום נקראת EISA (Expanded Industry Standart Bus), אשר מאפשרת פעולה בקצבים גבוהים יותר מאשר ISA BUS שפעולתו מוגבלת לקצבים של עד 20 MHz. המהירות הגבוהה של הערוץ אינה מאפשרת לכרטיסים שפותחו עבור ISA BUS לפעול במערכות DX, מכיון שאינם

עומדים במהירות העבודה של הערוץ. פירוט נוסף על נושאים אלה תמצא בפרקים אחרים בספר.

במחשבי PS/2 מחברי ההרחבה שונים לחלוטין. הם בנויים בארכיטקטורת **מיקרו ערוץ** (Micro Channel). למחברי מיקרו ערוץ **לא ניתן לחבר** כרטיסי תיאום שפותחו לערוצי ISA או EISA. גם במיקרו ערוץ קיימים מחברי הרחבה לכרטיסים ברוחב 8 סיביות, 16 סיביות, או 32 סיביות. במערכות אלו נמצא בדרך כלל 2.5 או 3 מחברי הרחבה, מכיון שמרבית ההתקנים נמצאים על גבי הלוח הראשי (מתאם מסך, מתאם מדפסת, מתאם טורי, מתאם ל-Joy Stick).

2.7.7 המעבד המתמטי

המעבד המתמטי הינו מעבד נוסף במערכת המחשב. הוא משמש לביצוע פעולות מתמטיות בלבד במספרים שלמים ובנקודה צפה במהירות גבוהה יותר מאשר המעבד הראשי. הוא פועל במקביל למעבד המרכזי וכך הוא יכול לשחרר אותו בזמן ביצוע פעולות מתמטיות לביצוע פעולות נוספות. אם המעבד המתמטי אינו מותקן במערכת משתמשים בתכנית מורכבת, אשר מדמה את פעולת המעבד המתמטי, לביצוע פעולות מתמטיות מורכבות שהמעבד המתמטי מבצע אותן בפעולה אחת.

את רכיב המעבד המתמטי מתקינים על הלוח הראשי במקום המיועד. המעבדים 8087, 80287, 80387SX, או 80387 צריכים להיות מותאמים לסוג המעבד הראשי לפי הפירוט שבהמשך.

מעבד מתמטי אינו חלק מתצורת המחשב ויש לרכוש אותו בנפרד (במחשבי 486 המעבד המתימטי נמצא בתוך המעבד הראשי). הכנסת מעבד מתמטי למערכת המחשב אינה משפרת את ביצועי המחשב בצורה מיידית ואוטומטית. עבור תכניות שאינן כוללות פקודות במספרים ממשיים, אין למעבד זה כל משמעות. מרבית התכניות היישומיות אינן משתמשות במעבד המתימטי גם אם הוא קיים במערכת. מכאן, שאם נרכוש ונכניס מעבד מתמטי למערכת המחשב ייתכן שביצועי המערכת לא ישתפרו כלל! תכניות מסוימות מחייבות שימוש במעבד מתמטי כתנאי הכרחי לביצוע התכנית, או לביצוע שלה במהירות סבירה. המעבד המתמטי מסוגל לבצע חישובים בנתונים ברוחב של עד 80 סיביות.

מחירו של המעבד מתימטי גבוה באופן יחסית למעבד המרכזי. התקנתו במערכת פשוטה וכל שנדרש הוא לשנות את מצב אחד המתגים על הלוח הראשי כדי לסמן למערכת כי הוא קיים. בחלק מהמערכות אסור להגדיר שקיים מעבד כזה, אם הוא לא מותקן, כי המערכת לא תוכל לפעול כלל.

המעבד המתמטי תומך במספר סוגים של משתנים על פי הטבלה הבאה:

סוג המשתנה	אורך המשתנה	תחום ערכי המשתנה
שלם (Word)	16	+32767 עד -32768
שלם (Short)	32	$+2X10^{9}$ עד $-2X10^{9}$
שלם (Long)	64	$+9X10^{18}$ עד $-9X10^{18}$
עשרוני ארוז	80	+999999999999999999 18 ספרות עד -999999999999999999
ממשי קצר	32	$\pm 3.37X10^{38}$ עד $\pm 8.43X10^{-37}$
ממשי ארוך	64	$\pm 1.6710^{308}$ עד $\pm 4.19X10^{-307}$
ממשי זמני	80	$\pm 1.2X10^{4932}$ עד $\pm 1.2X10^{-932}$

המעבד המתמטי מותאם לסוג מעבד מסוים בלבד (מעבד **80287** מותאם לסוגים אחדים). הטבלה הבאה מתארת את סוג המעבד, סוג המעבד המתמטי המתאים עבורו ומהירות הביצוע היחסית של מעבדים אלה:

מעבד 80486	מעבד מתמטי קיים במעבד	↑
מעבד 80386	מעבד מתמטי 80287, 80387	
מעבד 80386SX	מעבד מתמטי 80387SX, 80287	
מעבד 80286	מעבד מתמטי 80287	
מעבד 8086 / 8088	מעבד מתמטי 8087	ביצועים

2.8 אפיונים של סוגי מחשבים נפוצים

הטבלה הבאה מפרטת תכונות של סוגי מחשבים שונים הקיימים בשוק. היא מציגה את המחשבים על פי סוג המעבד שנמצא בהם ומפרטת את מירב התכונות אשר ניתן למצוא על לוחות המחשבים השונים. מן הראוי להדגיש שלא בכל לוחות המחשבים המבוססים על אותו מעבד נמצא את כל התכונות המוזכרות.

תכונה \ סוג מעבד	8088	8086	80286	80386SX	80386DX	80486
רוחב פס נתונים	8	16	16	16	32	32
רוחב פס כתובות	20	20	24	24	32	32
מעבד מתמטי	8087	8087	80287	80387SX	80387	פנימי
תדר עבודה MHz	5-12	5-12	6-20	16-20	16-40	25-50
אפשרות ל-OS/2	לא	לא	כן	כן	כן	כן
זיכרון RAM בסיסי	640KB	640KB	640KB	640KB	640KB	640KB
אפשרות חיבור זיכרון הרחבה Expanded מעל 640KB	ניתן לחבר	ניתן לחבר	ניתן לחבר זיכרון כזה אך אין זה כדאי כלכלית. במערכות אלו כדאי ומומלץ זיכרון מסוג Extended.			
	קיימים לוחות ראשיים מסוגים שונים אשר מכילים מעגל חומרה המאפשר מיתוג דפים. בחלק ממעגלים אלה ניתן למקם את הזיכרון גם מתחת 640K					
זיכרון Extended מירבי בלוח הראשי	אין	אין	עד 16MB	עד 16MB	עד 4GB(תיאורטי) (4-16MB)	עד 4GB (4-16MB)
זיכרון מטמון	אין	אין	אין	יש	יש	יש פנימי
סוגי כונני דיסקטים	5.25" - 360KB 3.5" - 720KB		5.25" - 360KB 1.2 MByte 3.5" - 720KB 1.44 MByte			
שעון זמן אמת Real Time	תוספת למערכת ולא חלק בסיסי		שעון זמן אמת נמצא כתקן בסיסי על גבי הלוח הראשי			
InterLeaved ו-Paged Memory	לא	לא	כן	כן	כן	כן
Shadow Ram	לא	לא	כן	כן	כן	כן
מחברי הרחבה	8	8/16	8/16	8/16	8/16/32	8/16/32
קביעת תצורת מערכת (Setup)	קביעה חלקית על ידי מתגים על הלוח ראשי		ברוב המערכות כיום גישה למשתמש בתוכנה אפשרית בשלב ה-BOOT. במערכות יבמ נעשה ה-Setup באמצעות דיסקט מיוחד.			

כן - משמעות: ניתן לחבר או למצוא לוחות מחשב המכילים את התכונה הנזכרת.

לא - משמעות: לא ניתן למצוא את התכונה בלוחות ראשיים אלה.

(ערך) - הגדלים המעשיים הקיימים בפועל בלוחות ראשיים מקובלים.

השוואת תכונות מחשבים על פי המעבד המרכזי

2.9 ספק המתח במערכת המחשב

ספק המתח של המחשב האישי מספק לו את המתחים השונים הנדרשים להפעלת הלוח הראשי וכרטיסי התיאום והבקרה. ספק המתח מופעל על ידי מתג הפעלה ON/OFF המשמש לשני תפקידים:

* הדלקת מערכת המחשב. כאשר הלוח הראשי מקבל מספק המתח רמת מתח של +5V מתחיל תהליך האתחול של המערכת.

* בתוך ספק המתח קיימת מערכת הגנה באמצעות נתיך (Fuse) חצי אוטומטי. כלומר, כאשר אחד ממתחי המוצא של הספק יצרוך זרם גבוה מדי, כמו במקרה של קצר, מערכת הנתיך תזהה זאת ותפסיק את אספקת כל המתחים למחשב ותעצור את פעולתו. נעילת המתחים במחשב אינה מלווה בנפילת מתג ההפעלה. כדי להחזיר את המצב לקדמותו, לשחרר את מערכת הנתיך ולהפעיל את המחשב, יש צורך לסגור את מתג ההפעלה ולהדליק את מערכת המחשב מחדש.

אזהרה - אין לפתוח את מכסה הספק ואין לטפל בחלקיו הפנימיים. המתחים בתוך קופסת הספק גבוהים ביותר **ומסוכנים לחיי אדם יותר ממתח רשת החשמל**. הסבר עיקרון הפעולה של הספק המובא כאן הינו לצרכי ידע כללי בלבד ולא לצורכי תיקון הספק!

2.9.1 עיקרון הפעולה

הספק במערכת המחשב האישי הינו **ספק ממותג**. עקרון הפעולה של ספק ממותג (ראה תרשים) מאפשר לבנות ספק מתח קטן במימדים ובמשקל מבלי לפגוע בכושרו לאספקה של זרמים גבוהים ביותר! המשקל הקטן מושג על ידי המבנה המיוחד של המעגלים הפנימיים.

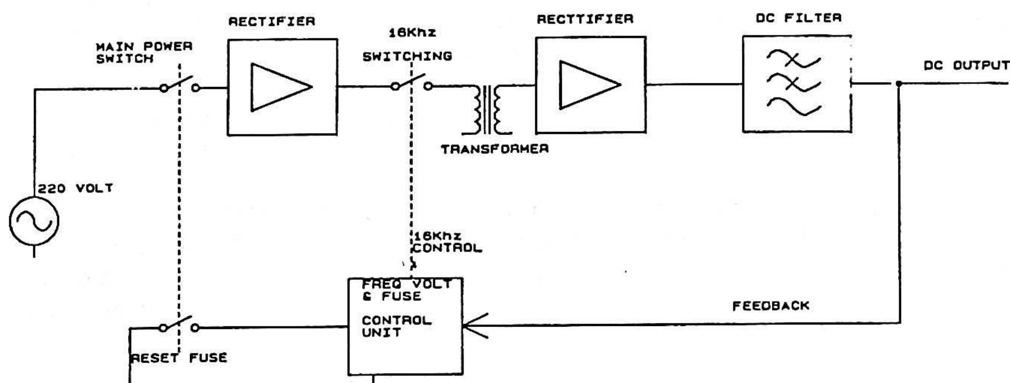
ספק מתח רגיל שאיננו ממותג, בנוי משנאי (Transformator) שהינו סליל גדול עם ליבת ברזל בתוכו, המשנה את מתח הכניסה של הרשת (220V) למתח ישר הנדרש במוצא הספק. הספק ממיר את מתח הכניסה למתח היציאה הרצוי באמצעות מעגלי יישור וסינון המספקים מתח מוצא יציב. בכדי ששנאי הפועל בתדר 50Hz המסופק על ידי חברת החשמל, יספק את המתחים הנדרשים במערכת המחשב ובזרמים הגבוהים הנדרשים בו, הוא צריך להיות גדול פי שלושה מגודל ספק ממותג ולשקול כשני מחשבים. מכיון שגודל ומשקל אלה אינם מעשיים לצרכים של מחשבים שולחניים, משתמשים בספק ממותג.

בספק ממותג כניסת המתח היא 220V/50Hz והוא מיושר כבר בכניסה של ספק המתח. כלומר, נקבל מיד ביציאה של המיישר הראשוני מתח ישר ברמה של 380V (נכון, 380V). מתח זה מתחבר לשנאי באמצעות מעגל מיתוג בתדר של 16KHz. שנאי הפועל בתדר כניסה של 16KHz הינו שנאי קטן יחסית לשנאי באותו יחס ליפופים העובד בתדר של 50Hz ובזה יתרון ההמרה ל-16KHz.

מערכת פיצוי העומסים השונים בספק ומערכת הנתיך החצי אוטומטי נכנסות כמערכות שליטה ובקרה על מעגל מיתוג תדר ה-16KHz. לדוגמה, כאשר יש

צורך בהפעלת הנתיך (כתוצאה מצריכת זרם מוגברת), מנתקים את תדר המיתוג של ה-16KHz בכניסה לשנאי ומפסיקים את כל מתחי המוצא של הספק.

ברור מאליו שגם מעגלי הסינון והיישור לתדר של 16KHz קטנים יותר ממעגלי יישור וסינון ל-50Hz, מכיון שהקבלים הנדרשים לצורך הסינון הינם קטנים יותר מאשר קבלים הנדרשים לסינון של 50Hz. ספק המתח מבקר את תקינות רמת המתחים השונים במערכת ומספק ללוח הראשי במחשב אות המציין את תקינות מתחי המוצא של הספק.



ספק מתח - תרשים מלבני

סוג הספק	הספק בוואט	זרם +5V	זרם -5V	זרם +12V	זרם -12V
ספק רגיל XT	130	15A	0.3A	4.2A	0.3A
ספק AT	200	20A	0.5A	8A	0.5A

זרמים ומתחים בספקים נפוצים

2.9.2 חיבורים חיצוניים

ספק המתח מתחבר אל הלוח הראשי וכונני הדיסקים/דיסקטים השונים באמצעות 5 או 6 פתילים היוצאים מהספק.

המתח מגיע מספק המתח של המחשב באמצעות שני פתילים המספקים ללוח הראשי את כל מתחי הפעולה הדרושים לפעולתו. באמצעות מחברי הרחבה הוא מעביר מתחים אלה למתאמים השונים המחוברים למערכת.

		צבע נפוץ	שימוש
1	∞	לבן	מסמן תקינות מתחים של הספק
2	∞	אדום	אות בקרה (Key) בשימוש בלוח ראשי של AT ומעלה
3	∞	צהוב	+12V
4	∞	כחול/חום	-12V
5	∞	שחור	Ground אדמה
6	∞	שחור	Ground אדמה

1	∞	שחור	Ground אדמה
2	∞	שחור	Ground אדמה
3	∞	כחול/לבן	-5V
4	∞	אדום	+5V
5	∞	אדום	+5V
6	∞	אדום	+5V

הצד האחורי של הלוח הראשי (הצד בו מתחבר לוח המקשים)

חיבורי מתח לכוננים

המתח לכוננים מגיע באמצעות פתיל המספק מתחים של +5V ו-+12V בזרמים גבוהים.

1	∞	צהוב	+12V
2	∞	שחור	Ground אדמה
3	∞	שחור	Ground אדמה
4	∞	אדום	+5V

פרק 3

מבנה מערכת ההפעלה ותהליך BOOT במחשב האישי

התפקיד העיקרי של מערכת ההפעלה הינו לספק למשתמש ממשק נוח וקל לפעולה מול מעגלי החומרה וההתקנים ההיקפיים המרכיבים את מערכת המחשב. מערכת ההפעלה השולטת בשוק המחשבים האישיים היא **DOS** (Disk Operating System) בהמשך נסביר בקצרה את מערכת OS/2 שפותחה על ידי יבמ עבור המחשבים האישיים מסדרת PS/2.

בפרק זה נעסוק בנושאים הבאים:

* הסבר של התהליך המתרחש במחשב מרגע הדלקתו ועד לרגע שבו מופיע **הסימן המנחה** (Prompt) של מערכת ההפעלה. הסימן המנחה מורה למשתמש כי ניתן להתחיל לבצע תכניות.

* פירוט המודולים והתכניות השונים המרכיבים את מערכת ההפעלה, תפקידיהם, שמותיהם והדרך שבה הם משתלבים זה בזה.

* פירוט התהליך שבו מבצעת מערכת ההפעלה את תכנית המשתמש.

בפרק זה נלמד את הפקודות השונות של מערכת ההפעלה DOS אשר נדרשות לתפעול של מערכת המחשב. לנוחות הקורא ניתן למצוא בנספח את תקציר כל הפקודות.

3.1 הגדרות

* **תהליך אתחול - Boot Process**
התהליך מתחיל בהדלקת המחשב או בלחיצה על Reset ומסתיים כאשר מופיע הסימן המנחה של מערכת ההפעלה על גבי המסך. כך מסמנת מערכת ההפעלה למשתמש שהמחשב מוכן לפעולה.

* **רשומת אתחול – BOOT Record**
רשומה, או תכנית אתחול, שנמצאת **בכל** דיסק, או דיסקט המכיל את מערכת ההפעלה. מקומה בגיזרה הראשונה (sector) של הדיסק/דיסקט ואורכה 512 תווים.

* **דיסק/דיסקט האתחול – Boot Disk/Diskette**
התוכנה המרכיבה את מערכת ההפעלה נטענת מדיסק/דיסקט האתחול אל זיכרון RAM של המחשב. קובצי ההפעלה של מערכת ההפעלה, הכתובים בשפת

מכונה, נמצאים גם הם בדיסקט/דיסק האתחול. אם הם לא נמצאים בו, אין הוא יכול לשמש למטרת אתחול. כדי לטעון את קובצי ההפעלה על דיסקט שאינו משמש כדיסקט אתחול יש להשתמש בפקודות FORMAT או SYS.

* **תכניות שירות של המערכת** (System Calls)
תכניות השירות הן חלק ממערכת ההפעלה ונמצאות דרך קבע בזיכרון המחשב כאשר הוא מופעל. הן נותנות שירותי הפעלה (System) לתכניות היישום ולמערכת ההפעלה עצמה. **אין** להחליף בין תכניות שירות אלו לבין תכניות שירות **חיצוניות** (External Commands) או תכניות שירות **פנימיות** (Internal Commands), שתוסברנה בהמשך.

* **פקודות פנימיות** (Internal Commands)
פקודות כמו DIR, שנמצאות דרך קבע בזיכרון המחשב כחלק של תכנית מערכת ההפעלה ואין צורך לקרוא אותן מדיסקט או דיסק לשם ביצוע.

* **פקודות חיצוניות** (External Commands)
בדיסקט האתחול נמצאים קובצי ההפעלה של DOS ובנוסף להם גם קבצים המכילים תכניות שהן פקודות שניתן לקרוא להן לביצוע. אלו הן פקודות חיצוניות, או תכניות שירות חיצוניות.

* **תכניות יישום** (Application Programs)
תכניות יישום הין תכניות שהמשתמש פיתח ומפעיל כדי לבצע משימות שונות. תכנית יישום יכולה להיות גם תכנית מוכנה, כמו מעבד תמלילים, או גיליון אלקטרוני. היא גם יכולה להיות תכנית שירות כמו עורך קובצי טקסט (Editor), מהדר (Compiler), מחולל יישומים וכו'.

3.2 מהי מערכת הפעלה?

מערכת הפעלה הינה אוסף של פקודות ותכניות שירות שנועדו לנהל את תהליך העבודה וזרימת המידע במחשב. מערכת ההפעלה נועדה לאפשר גם למשתמש ההדיוט להשתמש במחשב באופן פשוט ויעיל. היא חוסכת ממנו את הצורך להכיר את המבנה הפיסי של המחשב שבו הוא משתמש, את סוגי המדפסות, כונני הדיסקים ושאר הרכיבים וההתקנים.

מערכת ההפעלה מספקת למשתמש ממשק פקודות פשוט על מנת לבצע את הפעולות במערכת המחשב. היא מנהלת את מערכת הקבצים ומאפשרת שמירה מסודרת של הנתונים האגורים בהם. היא מפקחת על האופן שבו יבוצעו התכניות השונות ומאפשרת להן ליצור קשר פשוט ו"שקוף" עם התקני החומרה השונים זה מזה במבנה ובתכונות. הממשק שמציעה מערכת ההפעלה ניתן הן למתכנתים הכותבים ומפתחים תוכנה למערכת המחשב והן למשתמש הפשוט, אשר חסר כל רקע שהוא בתחום המחשבים.

מערכת ההפעלה הנפוצה ביותר במחשבים האישיים של יבמ והתואמים להם הינה מערכת ההפעלה DOS (Disk Operating System). מערכת ההפעלה מורכבת ממספר חלקים, אשר יוצרים יחדיו את המכלול השלם הנקרא "מערכת הפעלה" ומאפשרים למשתמש לבצע את פעילותו בצורה טובה, יעילה ופשוטה. מחשב אשר

נקנה בחנות מכיל אך ורק את מערכות החומרה השונות שהוזכרו בפרקים קודמים ואין בתוכו ממשק מתאים לצורך דו-שיח ושירותים למשתמש.

למערכת ההפעלה DOS מספר גירסאות. הגירסה הקיימת בעת כתיבת הספר הינה גירסה 5. הגירסאות שונות זו מזו, אך הן שומרות על תאימות כלפי מטה בלבד. כלומר, גירסה מתקדמת של מערכת ההפעלה תכיר את תכונות הגירסאות הנמוכות יותר ותוכל לפעול על דיסק/דיסקט מגירסאות קודמות. מצב הפוך לא מתקיים.

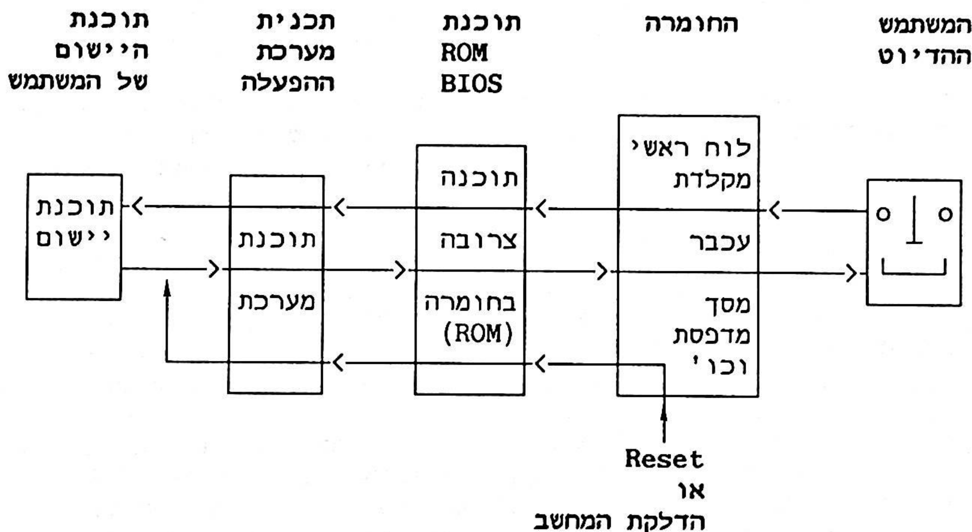
ההבדלים בגירסאות השונות של מערכת ההפעלה שונים ומגוונים. החל בתכניות שירות מקיפות ומעודכנות יותר, דרכי טיפול בדיסקים ובמסכים בצורה יעילה/מתוחכמת יותר ושיפור ביצועים של מערכת המחשב. התופעה השכיחה ביותר בגירסאות מתקדמות היא הכנסת שירותים נוספים למערכת ההפעלה. שירותים חדשים ונוספים אלה "מאומצים" משירותים שניתנים באמצעות תכניות שירות חיצוניות וכל משתמש "לא יכול לחיות בלעדיהם". נזכיר את Cache, DosShell, אישורי מחיקה לקבצים ועוד.

תוספת של שירותים למערכת ההפעלה גורמת לכך שגירסאות גבוהות יותר של מערכת ההפעלה תופסות מקום רב יותר בזיכרון מגירסאות נמוכות יותר. מגוון השירותים העדכניים יותר והממשק הנוח יותר למשתמש (כמו DosShell בגירסה 4 ומעלה), הופכים את המעבר לגירסה גבוהה יותר לאטרקטיבי יותר עבור המשתמש. בספר נתייחס לפקודות DOS בגירסה 4 ואילך המקנות למשתמש כלים יעילים יותר לטיפול במערכת המחשב.

גירסה 5, אשר יצאה לשוק בחודש יוני 1991, הינה הגירסה העדכנית ביותר של מערכת ההפעלה DOS. גירסה זו מיועדת לפעול על **מחשבי XT ומעלה**, אך משתמשי XT לא יוכלו ליהנות מכל שירותיה. מערכת ההפעלה גירסה 5 תספק למשתמש שירותים רחבים יותר בממשק ידידותי יותר מגירסאות קודמות. התוספות כוללות:

* יותר זיכרון פנוי למשתמש מתחת ל-640KByte. מערכת ההפעלה נטענת לזיכרון ההרחבה ולא לזיכרון הבסיסי.
* תכנית Dos Shell מתקדמת וידידותית יותר.
* הודעות שגיאה מפורטות ומדויקות יותר בצירוף מסכי עזרה למשתמש.
* היחלצות ממחיקה מוטעית של קבצים וביצוע עריכה - FORMAT - לדיסק באמצעות הפקודות UNDELETE ו-UNFORMAT.
* אפשרות לראות ולבצע את הפקודות האחרונות שניתנו ל-DOS על ידי המשתמש (DOSKEY).

תהליך האתחול (BOOT) של מערכת המחשב מתחיל בהדלקת המחשב או בלחיצה של המשתמש על לחיץ Reset הנמצא בחזית. בסוף התהליך, מערכת ההפעלה תופסת את מקומה בזיכרון ומתחילה לתפעל את מערכת החומרה מול המשתמש.



תרשים זרימה של הוראות המפעיל במחשב

3.2.1 מה עושה מערכת ההפעלה?

למערכת ההפעלה מספר תפקידים. התפקידים העיקריים של מערכת ההפעלה מחולקים על פי שלבי הביצוע שלהם:

* **שלב 1:** תהליך אתחול (BOOT). שלב זה כולל את הבאת המחשב למצב עבודה לאחר שמפעיל המחשב הסיט את מתג המתח, או ביצע Reset למחשב.

* **שלב 2:** תקשורת בין המפעיל למערכת החומרה במחשב ושליטה בסביבת העבודה של המערכת. המערכת מאפשרת ניהול החומרה, הקמת ספריות חדשות, מעבר מספריה לספריה, מעבר בין דיסקים, העתקת קבצים, מחיקת קבצים, ביטול ספריות וכו'.

* **שלב 3:** טעינה והרצה של תכניות יישום ותכניות מערכת ההפעלה עבור המשתמש. שלב טעינת והרצת תכניות משתמש נראה טבעי ופשוט, אך זהו שלב מורכב ביותר. הוא מתחיל כאשר המשתמש מקיש את שם קובץ התכנית, אשר מורה למערכת ההפעלה לקיים את פעולת **"טעינה וביצוע"**, או Load & Exec. פעולה זו מתבצעת בשלבים:

א. מערכת ההפעלה DOS **מחפשת** אם קיים קובץ בשם זה. החיפוש נעשה בספריה הנוכחית ואחר כך בספריות שהוגדרו בפקודה PATH. שמו של הקובץ חייב להסתיים בסיומת BAT, בסיומת COM או בסיומת EXE. אם קיים יותר מקובץ אחד בשם זה, סדר העדיפות הינו לפי הסיומת: Com, Exe, Bat (קובץ עם סיומת Com מתבצע ראשון).

ב. מערכת ההפעלה DOS **טוענת** את הקובץ מהדיסק/דיסקט לזיכרון RAM (פעולת Load). אם הקובץ הוא בעל סיומת BAT, היא תקרא אותו שורה אחרי שורה ותתייחס לכל שורה כאילו הוכנסה כפקודה על ידי

המפעיל (תהליך ביצוע של קובצי אצווה BATCH). כלומר, עבור כל שורה בקובץ נחזור לבצע את שלב 3 מתחילתו.

ג. מערכת ההפעלה DOS מכניסה לאוגרי המעבד IP ו-CS את הכתובת של תחילת התכנית שנטענה לזיכרון RAM בשלב ב'. כך, **המעבד מתחיל לבצע** את התכנית שביקש המשתמש (פעולת EXEC).

ד. מערכת ההפעלה מספקת שירותים לתכנית היישום שמתבצעת במחשב. בשלב זה השליטה במעבד היא בידי תכנית היישום. עם זאת, כאשר היא צריכה להשתמש בהתקן חומרה כלשהו כדי לקרוא רשומה מדיסקט, להקרין למסך, לקרוא ממקלדת וכו', היא פונה למערכת ההפעלה לקבל את השירות הזה. לאחר מתן השירות מחזירה מערכת ההפעלה את השליטה במעבד לתכנית היישום להמשך הביצוע.

ה. מערכת ההפעלה מורידה בסיום הביצוע את תכנית היישום מזיכרון RAM וחוזרת לשלב השני (Prompt). שלב זה בעייתי מבחינת תפעול ועל כן נמצא שברוב המקרים שבהם המחשב "נתקע" ומפסיק לפעול הוא נמצא בשלב זה. מערכת ההפעלה אינה מגבילה את תכנית היישום. המתכנת מקבל אמצעים להפעלת המערכת ללא בקרה כלשהי ועל כן פעולות לא מסודרות ולא חוקיות שהוא מבצע עלולות לגרום להפסקת פעולת המחשב.

במערכות ההפעלה הפועלות במחשבים המשרתים משתמשים רבים בעת ובעונה אחת לא ניתן למשתמש/מתכנת מרחב תמרון כזה, ואם הוא שגה – רק התכנית שלו מופסקת.
הסברים מפורטים אודות שלב הטעינה ניתן למצוא בפרק 6.

3.3 המרכיבים העיקריים של מערכת ההפעלה

3.3.1 רשומת האתחול (BOOT Record)

רשומת האתחול (Boot) נמצאת בכל דיסקט או דיסק מגנטי המכילים את מערכת ההפעלה DOS. היא "תופסת" תמיד **גיזרה** (Sector) אחת, שגודלה 512 בתים. מקומה במסלול 0, גזרה 1, צד 0 שבכל דיסקט שעבר תהליך של הכנה לעבודה על ידי תכנית FORMAT הכלולה בתכניות השירות של מערכת ההפעלה.

רשומת האתחול מאפשרת למחשב להבחין אם הדיסקט שנמצא במערכת מכיל את התכניות של מערכת ההפעלה DOS. כדי לאפשר פעולה התחלתית של מערכת ההפעלה, חייב הדיסקט/דיסק להכיל מספר קבצים הקרויים **גרעין** (Kernel). אם הדיסקט שבכונן A (או הדיסק בכונן C, אם לא הכנסנו דיסקט לכונן A) אינם מכילים את גרעין מערכת הפעלה המאפשר תחילת פעולה במחשב, תופיע הודעת שגיאה על המסך בזמן האתחול: Non System Disk.

בדיסקים קשיחים נמצא את רשומת האתחול בגיזרה הראשונה על גבי מחיצת DOS (DOS Partition, ראה הסבר בהמשך). גיזרה זו אינה חייבת להיות הראשונה על גבי הדיסק. בגיזרה הראשונה בדיסק נמצאת טבלת מחיצות (Partition table). התכנית הנמצאת בתחילת הגיזרה תחפש בטבלה זו את

מחיצת DOS ותחליט לאיזו רשומת BOOT יש להעביר פיקוד (הסבר מפורט תמצא בפרק הדן במערכות הדיסקים במחשב).

3.3.2 קובץ ממשק (Interface) לתכניות ROM

תכניות ROM BIOS הן חלק מחומרת המחשב ו"צרובות" בחומרה. הן מבצעות פעולות בסיסיות של המערכת, כמו גישה להתקני חומרה מסוגים ובעלי תכונות שונים (דיסקים קשיחים, כונני דיסקטים, מסכים, מדפסות ועוד). תכניות אלו מאפשרות למתכנת לפנות אל התקני החומרה הקשורים למערכת ולבצע בהם פעולות שונות ללא צורך להכיר את מבנה החומרה וצורת הגישה הפיסית להתקנים אלה.

הקובץ מהווה ממשק בין התכניות הכתובות בזיכרון ROM BIOS המגיע יחד עם מעגלי החומרה של המחשב ומהווה חלק בלתי נפרד מהם, לבין הקובץ המהווה ממשק לתכניות השונות הפועלות במחשב. קובץ זה מכיל את התכנית SYSINIT שתפקידה ליצור את סביבת מערכת ההפעלה במחשב האישי.

תכנית SYSINIT אחראית להמשך תהליך טעינת מערכת ההפעלה כפי שמפורט בתהליך האתחול. התכנית בודקת את מרחב הזיכרון הקיים בפועל עבור מערכת ההפעלה, פותחת ומבצעת את הפקודות הרשומות בקובץ CONFIG.SYS שורה אחר שורה (פקודות כמו Files, Buffers) ומכינה מקום בזיכרון עבור טבלאות לטיפול בקבצים ושטחי עבודה לנתונים לקריאה וכתיבה מדיסקים (Buffers).

תכנית SYSINIT טוענת את **התוכנות להפעלת ההתקנים** (Device Drivers) אשר רשומות בקובץ CONFIG.SYS, בודקת את הכוננים השונים וההתקנים המחוברים למערכת ומבצעת אתחול של מעגלי החומרה (אם הדבר נדרש על ידי תכניות ההפעלה להתקנים). תכנית SYSINIT טוענת את הקובץ IBMDOS.COM ואת הקובץ COMMAND.COM.

תכנית SYSINIT הינה הקובץ IBMBIO.COM במערכת ההפעלה DOS של יבמ, ובמערכת ההפעלה של מיקרוסופט שם הקובץ הוא IO.SYS. קובץ זה חייב להיות הקובץ הראשון הקיים בספריית השורש (Root Directory) בדיסק/דיסקט של מערכת ההפעלה. כאשר נקיש את הפקודה DIR לדיסק/דיסקט המכילים את מערכת ההפעלה שם קובץ זה לא יוצג מכיון שהוא מוסתר (Hidden).

3.3.3 קובץ תכניות שירות של מערכת ההפעלה DOS

קובץ תכניות השירות של מערכת ההפעלה כולל את השירותים הנדרשים עבור פסיקות השירות השונות. תכניות השירות מספקות למתכנת ולתכניות היישום השונות ממשק ברמה גבוהה, המאפשר גישה נוחה להתקנים השונים של המחשב, לניהול מערכת הקבצים ומערכת המחשב כולה. תכניות השירות מהירות וקלות לגישה. קריאה לתכנית שירות מתורגמת לקריאה או קריאות לקובץ הממשק לצורך השלמת הפעולה שנתבקשה. הגישה לתכניות השירות היא בעזרת מנגנון פסיקה ושימוש באוגרים השונים של המעבד. אוגרי המעבד משמשים להעברה של פרמטרים וכתובות של טבלאות שונות הנדרשות לביצוע תכנית השירות.

תכנית האתחול הנמצאת בקובץ זה נקראת על ידי תכנית SYSINIT. תכנית האתחול בונה את הטבלאות לשימוש מערכת ההפעלה, מאתחלת את וקטורי הפסיקה המתאימים לתכניות השירות, בודקת את מספר הדיסקים/כוננים המחוברים למערכת ומכינה שטחי עבודה בזיכרון להעברת נתונים בין מערכת הדיסקים לבין מערכת ההפעלה. תכנית זאת גם מציגה את גירסת מערכת ההפעלה.

במערכת הפעלה DOS של יבמ נקרא קובץ זה IBMDOS.COM ובמערכת DOS של מיקרוסופט נקרא הקובץ MSDOS.SYS. הקשה של הפקודה DIR בדיסקט המכיל גרעין של מערכת ההפעלה לא תציג קובץ זה, מכיון שהינו מוסתר (Hidden). קובץ זה חייב להיות הקובץ השני בדיסק/דיסקט לאחר הקובץ IBMBIO.COM.

3.3.4 מעבד הפקודות – Command Processor

תכנית מעבד הפקודות יוצרת ממשק לתקשורת בין המשתמש לבין מערכת ההפעלה. כל פקודה שנכתבת על ידי המשתמש נקלטת על ידי מעבד הפקודות ומבוצעת על ידו בשיטת **"טען ובצע"** (Load & Exec). מעבד הפקודות מכיל מספר מרכיבים המספקים ממשק לקליטת פקודות מהמשתמש, הרצה וביצוע של קובצי אצווה (Batch) אשר החשוב בהם הוא קובץ Autoexec.bat.

לתכנית המעבד יש שלושה חלקים:

חלק ראשון (Initialization Part)

חלק זה מכיל תכנית אתחול למעבד הפקודות וקוד לביצוע אוטומטי של הקובץ AUTOEXEC.BAT בסוף תהליך האתחול (BOOT). קובץ AUTOEXEC.BAT הוא קובץ אצווה המכיל פקודות DOS, אשר הוכנו על ידי המשתמש כדי להורות למערכת ההפעלה לבצע סדרה קבועה של פקודות בכל פעם שמדליקים את המחשב. ההרצה של פעולות אלו בסיום תהליך Boot מכוונת להכין את סביבת הפעולה של מערכת ההפעלה ולהתאימה לצורכי המשתמש (התקנת עברית, הגדרת ספריות ודרכי גישה אליהן, ביצוע של תכניות שירות כמו תכנית אנטי-וירוס, טעינת פונטים לבקר מסך VGA וכד').

חלק האתחול של תכנית COMMAND.COM טוען את שני החלקים הנוספים של מעבד הפקודות למקום פנוי בזיכרון RAM של המחשב ומשאיר אותם שם. בגמר פעולות אלו חלק האתחול אינו נשאר בזיכרון המחשב, כי אין בו צורך.

חלק שני (Resident Part)

החלק השני של מעבד הפקודות טעון **באופן קבוע** בחלק התחתון של זיכרון המחשב. זוהי **תכנית תושבת** (Resident program). החלק הקבוע של מעבד הפקודות מכיל קוד המאפשר לטעון את החלק הארעי (transient) של מעבד הפקודות (ראה להלן), מאפשר להחזיר שליטה למעבד הפקודות בסיום תכנית משתמש (פסיקה 22Hex של DOS), מטפל בלחיצות המשתמש על Control+Break (פסיקה 23Hex של DOS) ומטפל בפסיקה של שגיאות קריטיות בתכנית (פסיקה 24Hex של DOS).

חלק שלישי (Transient Part)

החלק הארעי של מעבד הפקודות מקשר בין המשתמש לבין מערכת ההפעלה. הקשר אל המשתמש מתקיים באמצעות הסימן המנחה (Prompt), אשר קורא את פקודות המשתמש מלוח המקשים ומפעיל תהליך הממלא את הבקשה/ההוראה.

החלק הארעי של מעבד הפקודות טעון בחלק הגבוה של זיכרון המחשב. תכנית יישום יכולה להסיר חלק זה מזיכרון המחשב ולהשתמש בו לצרכיה. בכך היא יכולה להגדיל את כמות הזיכרון העומד לרשותה. בסיום התכנית מחזיר החלק הקבוע של מעבד הפקודות את החלק הארעי לזיכרון, על מנת שניתן יהיה להמשיך בפעילות במחשב. החלק הנייד מכיל את הפונקציות הבאות:

1) פקודות פנימיות של מערכת ההפעלה DOS, אשר טעונות דרך קבע בזיכרון המחשב, כמו למשל DEL, DIR, COPY וכו'. התגובה שלהן מיידית, כי אין צורך לקרוא אותן מדיסק/דיסקט.

2) מעבד הפקודות של קובצי אצווה המאפשר ביצוע של הפקודות הרשומות בקבצים אלה.

אם המשתמש מבקש לבצע תכנית חיצונית של מערכת ההפעלה (פקודה מתוך ספריית התכניות של DOS), או תכנית יישום כלשהי, החלק הארעי של מעבד הפקודות COMMAND.COM קורא לתכנית שירות של DOS הנמצאת בקובץ תכניות השירות. תכנית השירות טוענת ומפעילה את התכנית המבוקשת (Load & Exec). כלומר, מעבד הפקודות ממלא את בקשת המשתמש על ידי ביצוע פקודה פנימית מתוך מעבד הפקודות הטעון בזיכרון המחשב, או על ידי ביצוע קובץ עם סיומת BAT, COM, או EXE מתוך הספריה שבה נמצא המשתמש באותו רגע, או לאחר חיפוש של הקובץ במסלול החיפוש שהוגדר באמצעות הפקודה PATH של מערכת ההפעלה (לפי שלב **3** של סעיף **3.2.1**). ראה הסברים נוספים בפרק **6**.

התכניות הטעונות באופן קבוע בחלק הזיכרון הגבוה של המחשב נקראות **גרעין** (Kernel) מערכת ההפעלה DOS, או אוסף הפקודות הפנימיות של מערכת ההפעלה. גודל גרעין מערכת ההפעלה משתנה כתלות בגירסאות השונות שלה. גודלו של הגרעין עולה ככל שעולה גירסת מערכת ההפעלה וקטן הזיכרון בפועל עבור המשתמש. לגרעין מערכת ההפעלה הוכנסו תכניות נבחרות שמשתמשים בהן בתכיפות גבוהה, על מנת לחסוך בזמן הטעינה של תכניות אלו מהדיסק/דיסקט. גודלו של הגרעין כ-50KByte, שאר תכניות DOS תופסות כמה מאות KByte. בגירסה 5 ומעלה של DOS ניתן לטעון חלק זה לזיכרון ההרחבה.

את קובץ COMMAND.COM ניתן לראות בפקודת DIR של מערכת ההפעלה בכל דיסק/דיסקט שניתן לבצע דרכו תהליך BOOT תקין. פירוט של כל הפקודות הפנימיות הכלולות במעבד הפקודות ניתן למצוא בנספח פקודות DOS.

3.3.5 תכניות שירות חיצוניות (External Commands)

תכניות שירות מסוגים שונים נועדו להשלים את מגוון התכניות השייכות לגרעין מערכת ההפעלה הקיים במעבד הפקודות. נזכיר כמה מן השימושיות ביותר כמו DISKCOPY, PRINT, FORMAT ואחרות. תכניות השירות החיצוניות

מגיעות על גבי דיסקטים בקבצים נפרדים. שם הפקודה הינו כשם הקובץ המכיל אותה וסיומת הקובץ הינה COM או EXE.

כדי שאפשר יהיה לטעון תכנית חיצונית, היא צריכה להמצא במערכת המחשב בדיסקט או בדיסק באופן שניתן לגשת אליה, לטעון אותה לזיכרון ולבצע. בגמר ביצוע של כל תכנית כזאת מוחזרת השליטה למעבד הפקודות ההתושב בזיכרון המחשב (Resident) והוא אשר מחזיר את הסימן המנחה ומאפשר המשך פעולה. פירוט של הפקודות החיצוניות תמצא בנספח פקודות DOS.

3.3.6 תכניות הפעלה להתקנים - Device Drivers

תכניות ההפעלה להתקנים מאפשרות למשתמש להגדיר את התקני החומרה במערכת ולשנות אותם, להוסיף התקנים ולשפר את הביצועים של מערכת המחשב. לא ניתן להריץ תכניות אלו באופן רגיל על ידי הקשת פקודה למעבד הפקודות, אלא באמצעות קובץ CONFIG.SYS, שבו הן רשומות. במהלך BOOT תכניות אלו נטענות לזיכרון RAM של המחשב ונמצאות בו כל עוד הוא פועל. הסיומת שלהן היא SYS, DRV או BIN.

שם ההתקן (Device)	סוג ההתקן
Aux	כרטיס התקשורת הטורי Com1
Com1	כרטיס מתאם טורי ראשון
Com2	כרטיס מתאם טורי שני
Com3	כרטיס מתאם טורי שלישי
Com4	כרטיס מתאם טורי רביעי
Con	המקלדת כקלט, או המסך כפלט
Lpt1	כרטיס מתאם מקבילי ראשון
Lpt2	כרטיס מתאם מקבילי שני
Lpt3	כרטיס מתאם מקבילי שלישי
Nul	התקן דמה. משתמשים בו כאשר לא רוצים שפלט מהתכנית ייכתב במסך או במדיה מגנטית

התקני חומרה תקניים ב-DOS

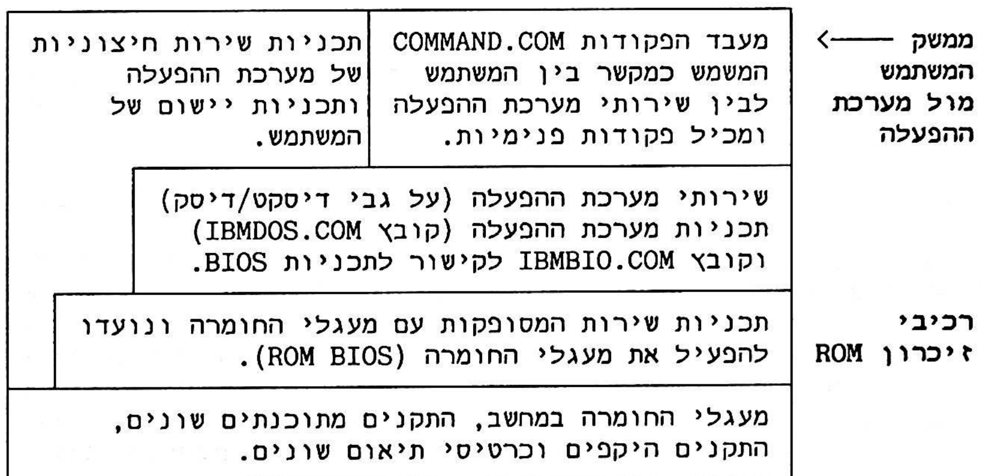
תכניות הפעלה אלו הן מסוג Character Device Drivers ודומות מבחינת ההתנהגות לקובץ. כלומר, אפשר לקרוא ולכתוב מתוכן. ישנן תכניות מסוג Block Device Drivers שמתנהגות כמו כונן או ספריה (לדוגמה, חלוקה למחיצות בדיסק קשיח). תכניות ההפעלה להתקנים מסופקות יחד עם מערכת DOS, או על ידי יצרני ההתקנים. ניתן לכתוב תכניות הפעלה להתקנים ייחודיים או לכתוב תכניות אחרות להתקנים הקיימים.

3.4 מבנה מערכת ההפעלה DOS

התרשים הבא מתאר את מרכיבי מערכת ההפעלה, במבנה של רמות, אשר כל אחת מהן מופעלת מהרמה שמעליה ומקבלת שירותים מהרמה שמתחתיה. שירותים אלה ניתנים על ידי שימוש בפקודות מתאימות. מסלול גישה זה הינו לעתים ארוך מדי מבחינת זמן הביצוע ועל כל ניתן גם לקצר אותו על ידי פניה ישירה מרמה אחת לרמה הרחוקה ממנה בהררכיה, וכך לקצר את משך הביצוע של התכנית. החיסרון הוא בכך שהדבר חורג מהסטנדרט שנקבע על ידי יצרני מערכת ההפעלה והוא מתאים לגירסה מסוימת של תוכנה וחומרה בלבד.

יצרני מערכת ההפעלה אינם ממליצים על גישה כזו, אלא על פעולה היררכית ברמות, המבטיחה תאימות של תוכנה גם בשינוי גירסאות של מערכות התוכנה והחומרה במערכת. למרות האמור לעיל, נמצא תכניות הפועלות ישירות מול חומרת המחשב או מול תוכנת ROM BIOS בשל יתרון הזמן העצום על פני פעולה באמצעות שירותי מערכת ההפעלה.

תכנית ממשק המשתמש פועלת מול תכנית מעבד הפקודות COMMAND.COM. היא קוראת את פקודות המשתמש מהמקלדת, טוענת ומריצה תכנית מבוקשת, או מבצעת את השירותים המבוקשים על ידו.



רמות מערכת הפעלה DOS

3.5 שירותי DOS לתכנית היישום

מערכת ההפעלה DOS מספקת שירותי מערכת לתכנית היישום (שלב 3ד בסעיף 3.2.1) בעזרת מנגנון פסיקות (Interrupts). התכניות שמספקות שירותים אלה נקראות שירותי DOS (DOS Services).

3.5.1 השימוש בפסיקות במערכת ההפעלה

במעבדי 8086 ניתן להגדיר 256 פסיקות שונות. פסיקות אלה ניתנות להפעלה גם בחומרה וגם בתוכנה.

*** מדוע צריך פסיקות חומרה/תוכנה?**

נניח שהתקן חיצוני, כמו למשל המקלדת, רוצה לשלוח נתונים למחשב. לשם כך המעבד צריך להפסיק את המשחמה שהוא עסוק בה, ביצוע תכנית, לקבל את הנתונים ולחזור ולהמשיך בביצוע התכנית ממקום שנמצא בו לפני הפסיקה. המקלדת צריכה להודיע למעבד על "רצונה" להזין נתונים והמעבד צריך לפנות אליה כדי לקבל אותם. בלעדי פעולות אלו לא ניתן לבצע את קליטת הנתונים מהמקלדת.

ההפרעה של המקלדת למעבד במהלך פעולתו הינה **פסיקה** (Interrupt). את תהליך הפסיקה והתקשורת למעבד משיגים באמצעות פסיקת חומרה. פסיקות תוכנה משמשות במערכת ההפעלה כמנגנון המאפשר ניידות של תכניות השירות בזיכרון.

*** מדוע צריך ניידות וגמישות של תכניות שירות?**

נניח שקיימות שתי תכניות שירות, אחת שמטפלת בקליטת נתונים מהמקלדת ואחריה תכנית שמטפלת בהצגת תווים על המסך. אם רוצים לשפר ולהרחיב את התכנית הראשונה, עשוי הדבר לגרום לשינוי בכתובת ההתחלתית של התכנית השניה. כל תכנית שהתייחסה לכתובת ההתחלתית של התכנית השניה לא תוכל לפעול יותר במערכת המחשב. הפתרון מושג על ידי שימוש במנגנון פסיקות.

*** כיצד משתמשים בפסיקות לקבלת ניידות בתכניות?**

וקטור הפסיקות (שמקומו קבוע בחומרה בין כתובת 0 ל-1K בזיכרון RAM) הינו טבלה המכילה את כתובת תכניות השירות השונות. התוכנות נטענות לזיכרון בזו אחר זו וכתובותיהן מוכנסות לטבלת ווקטור הפסיקות.

כאשר תכנית יישום זקוקה לשירות מסוים, היא מפעילה את הפסיקה המתאימה לשירות הנדרש באמצעות מנגנון הפסיקה. אין היא יודעת היכן נמצאת תכנית השירות בזיכרון. תפקיד זה מוטל על המעבד אשר יגש לווקטור הפסיקות, יקח ממנו את כתובת תכנית השירות ויבצע אותה. שינויים והרחבות של תכניות השירות יחייבו עדכון של וקטור הפסיקות בלבד והתכניות יוכלו להתבצע באופן תקין גם לאחר ההרחבה.

3.5.2 שירותי DOS

המשימה החשובה ביותר בשירותי מערכת ההפעלה היא קריאה וכתיבה להתקנים. התקן (Device) הוא רכיב, או אביזר חומרה הקשור למחשב למטרת קלט ופלט. דוגמאות להתקנים השונים: מסך, לוח מקשים, מדפסת, דיסק, דיסקט, שעון פנימי, כרטיס תקשורת טורית וכד'. עבור התקנים בסיסיים מספקת מערכת ההפעלה שירותי כתיבה/קריאה מיוחדים. יצרני חומרה לא תקנית יכולים להשתמש בכלים של מערכת ההפעלה לתכנות תכניות הפעלה להתקנים (Device Drivers) אשר מותאמות לחומרה המיוחדת שהם מספקים.

מערכת ההפעלה מאפשרת לבצע שינויים בסביבת העבודה של המערכת מתוך תכנית היישום. שינויים אלה כוללים מעבר בין ספריות, מחיקה והוספה של ספריות וקבצים, שינוי מצב השעון הפנימי, ניהול זיכרון ועוד. מערכת ההפעלה מספקת כלים לניהול התוכנה כמו שירותי Overlay ושירותי זיכרון הרחבה.

3.5.3 כיצד מקבלת תכנית היישום את השירותים

תכניות השירות (DOS Services) מופעלות על ידי קריאה לפסיקת שירות. מערכת ההפעלה משתמשת בפסיקות רבות, אבל תכנית היישום אמורה לקרוא רק **לפסיקה 21Hex**, שהינה פסיקת השירות הראשית. סוג השירות המבוקש, או מספר השירות, מוכנס כמספר באוגר AX. בספרים רבים מוזכר רק מספר השירות והקורא אמור לדעת שמדובר במספר שיש להכניס לאוגר AX בקריאה לפסיקה 21Hex. יש לזכור כי את סוג השירות המבוקש והפרמטרים השונים לשירות יש להכניס לאוגרים המתאימים לפני הקריאה לפסיקת השירות של מערכת ההפעלה.

המהדרים של שפות התכנות מכניסים קוד מכונה שקורא לפסיקה 21Hex כאשר המתכנת מבקש קריאה/כתיבה להתקנים: למשל, בפקודות PRINT בבייסיק, WRITE בפסקל, PRINTF בשפת C. במהדרים יעודיים קיימת אפשרות לפעולה ישירה מול מנגנון פסיקות מערכת ההפעלה ותכניות BIOS (ראה תכנית בנספח "תכניות דוגמה").

מתכנתים המעוניינים לפנות ישירות לתכניות השירות, חייבים לעיין בספר DOS Technical Reference של יבמ, או בספר אחר המיועד למתכנתים ב-DOS.

3.6 תהליך האתחול במחשב

תהליך האיתחול (Boot) מתחיל בהדלקת המחשב, או בלחיצה על לחיץ Reset (הנמצא ברוב המחשבים הקיימים בשוק). התהליך מסתיים בדרך כלל לאחר מספר שניות בהצגת הסימן המנחה (Prompt) של מערכת ההפעלה A> או C>. במחשבי IBM PC/G ומחשבי IBM PC/XT ישנים התהליך נמשך כדקה, אך בדרך כלל מספר עשרות שניות.

בתהליך האתחול מערכת ההפעלה נטענת לזיכרון המחשב והמשתמש מקבל אפשרות לפעול באמצעות הממשק שמספק מעבד הפקודות. התהליך מורכב ממספר שלבים,

אשר מאפשרים למערכת ההפעלה "לתפוס פיקוד" על מערכת החומרה של המחשב. ניתן להבחין בתהליך זה בשני שלבים עיקריים השונים במהותם (ראה תרשים זרימה של תהליך האתחול):

3.6.1 שלב 1 - בדיקה כללית של המחשב

בשלב זה מתבצעת תכנית הכתובה ב-ROM BIOS של המחשב. היא מבצעת בדיקה כללית של מעגלי החומרה של המחשב ומעבירה את השליטה לתכנית אתחול של DOS, אם היא נמצאת בכונן המחשב. על שלב זה אחראית החומרה והוא מתבצע ללא קשר למערכת ההפעלה. כאשר מדליקים את המחשב או מבצעים Reset, מתחיל המעבד המרכזי להריץ את תכנית האתחול של ROM BIOS מהכתובת 0FFFF:0000Hex (ראה הסבר בסעיף Reset של המעבד בפרק 1).

תכנית ROM BIOS מכילה שגרות לבדיקת שלימות של רכיבי החומרה השונים והכנתם לפעולה תקינה. רכיבי חומרה הקיימים היום מחייבים תכנות הקובע את אופן הפעולה שלהם והוא מתבצע בכל הדלקה או קביעה מחדש (Reset) של המחשב, על מנת שיתחילו לפעול באופן הנדרש מהם. הבדיקה כוללת את המעגלים הבאים: זיכרון ראשי, לוח ראשי וכל הרכיבים אשר עליו (אתחול של השעון 8253, בקר פסיקות 8259, בקר DMA 8237), מקלדת, מסך, כונני דיסקים/דיסקטים, מתאמי מדפסות, מתאמים טוריים וכו'. בתהליך זה מעודכנים הווקטורים של פסיקות השירות השונות של תכנית ROM BIOS.

פירוט של פסיקות תוכנה וחומרה ניתן לקבל על ידי תכניות שירות כמו SysInfo של Norton, תכנית CHECHIT לבדיקת התקני חומרה ותכניות אחרות.

לאחר אתחול הרכיבים בלוח הראשי מחפשת תוכנת BIOS הראשית תכניות ROM BIOS נוספות. כרטיסי הרחבה של דיסקים קשיחים, רשתות תקשורת ומסכים מסוימים מכילים תכניות BIOS ברכיבי ROM הקיימים על כרטיס ההרחבה. רכיבי זיכרון אלה ממופים בכתובות מעל כתובת C0000Hex והתכניות מזוהות על ידי המלה AA55Hex בתחילת בלוק של 2K בזיכרון.

בשלב זה, תכנית BIOS הראשית עוברת על כל הזיכרון בדילוג של 2K כדי לחפש את סימן הזיהוי AA55. אם נמצא זיהוי כזה, אוגר IP מקבל את הערך של הכתובת שאחרי הזיהוי וכך מקבל כרטיס ההרחבה את השליטה במעבד לצורך אתחול עצמי של הכרטיס. חובת כרטיס ההרחבה להחזיר שליטה לתכנית BIOS הראשית, כדי להמשיך את התהליך. עתה עוברת תכנית BIOS לשלב הבא.

בשלב בדיקת מערכת החומרה, כל תקלה שמאובחנת ומזוהה כקריטית להמשך תפקוד המערכת מדווחת באמצעות קוד שגיאה על גבי המסך וגורמת להפסקת התהליך. תקלה שאינה קריטית ואינה מונעת המשך תפקוד חלקי המערכת תדווח באמצעות קוד שגיאה, אך לא תעצור את תהליך BOOT.

קוד	הסבר קוד השגיאה
02x	תקלה בספק מתח או באחד המתחים השונים במערכת.
1xx	תקלה בלוח ראשי - System Board.
20x	תקלה בזיכרון ראשי RAM (בדרך כלל יופיע גם הקוד המסמן Parity Check).
30x	תקלה במקלדת.
6xx	תקלה בכונן דיסקטים (Floppy Drive).
17xx	תקלה בדיסק קשיח.

x - ערך מספרי כלשהו

קודי שגיאה בתהליך BOOT



השלבים בתהליך האתחול - BOOT

בגמר השלב מנסה תכנית BIOS לקרוא את רשומת BOOT מכונן A במערכת. אם בדיסקט שבכונן A קיימת רשומת BOOT, היא תוטען לזיכרון המחשב ותקבל פיקוד לצורך המשך ביצוע שלב 2. אם אין דיסקט בכונן, תכנית BIOS תנסה לטעון את רשומת BOOT ממחיצת DOS הפעילה בכונן הקשיח C ותעביר לרשומה זו שליטה לביצוע שלב 2 של BOOT.

אם יש דיסקט בכונן A, אך אין בו רשומת BOOT, או אם בדיסק הקשיח לא נמצאת רשומה כזו, תופיע על גבי המסך הודעת שגיאה המוסרת שאין דיסקט המכיל מערכת הפעלה על גבי המערכת. במערכות ישנות של יבמ היתה מועברת השליטה במקרה כזה לתוכנת ROM BASIC שנמצאה על הלוח ראשי של המחשב לתחילת עבודה בבייסיק. כיום, לא נמצא במחשבים הקיימים בשוק את רכיבי החומרה המכילים BASIC.

בגמר שלב 1 יודע המחשב אילו התקני חומרה יש במערכת ואם הם כשירים לפעולה, ואת הכתובת שבה הם נמצאים. הוא רושם פרטים אלה ב-RAM באיזור ידוע מראש. מערכת ההפעלה DOS משתמשת בנתונים אלה במהלך עבודתה.



שלב ראשון בתהליך BOOT

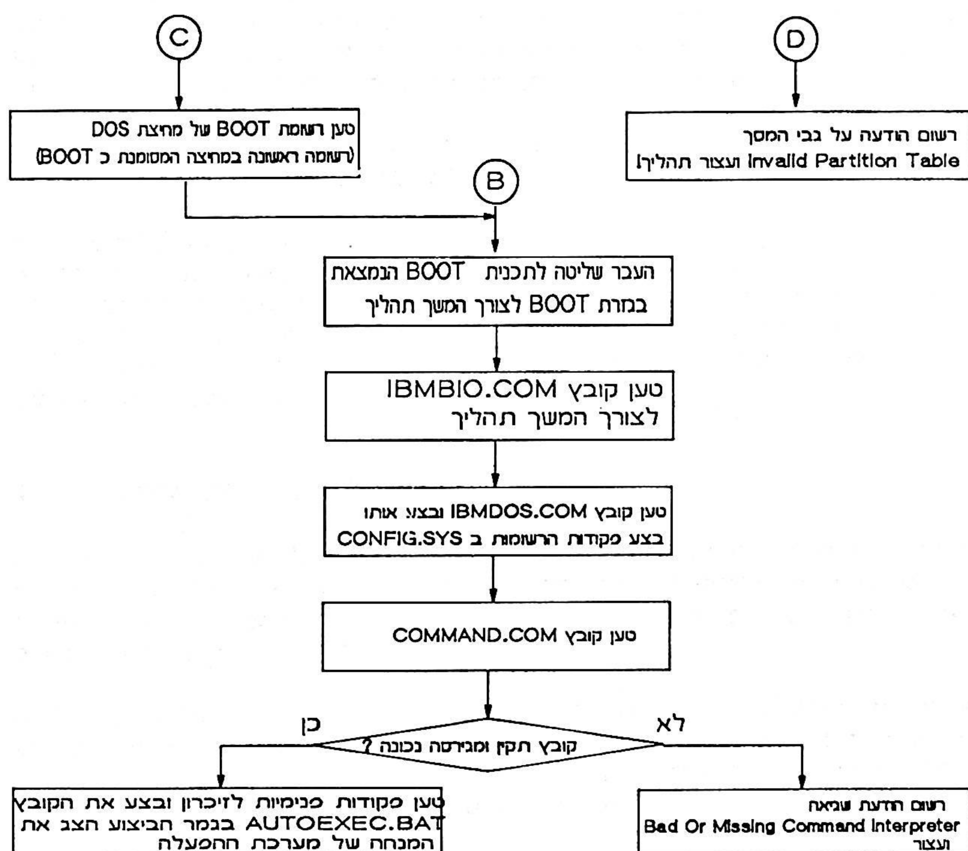
3.6.2 שלב 2 - טעינת מערכת ההפעלה לזיכרון המחשב

בשלב זה, התכנית שנקראה מגיזרת BOOT שבדיסקט/דיסק מקבלת שליטה ומתחילה להתבצע בטעינה של מערכת ההפעלה לזיכרון המחשב. מערכת הפעלה מתחילה לבצע סדרת בדיקות ולטעון תכניות ומסיימת בפניה אל המשתמש על ידי הסימן המנחה (Prompt).

תכנית BOOT שנטענה בודקת את ספריית השורש בדיסק/דיסקט ומוודאת ששני הקבצים המוסתרים (IBMBIO.COM ו-IBMDOS.COM, או IO.SYS ו-MSDOS.SYS) נמצאים ראשונים בספריית השורש וטוענת אותם לזיכרון. היא מעבירה את השליטה לתכנית האתחול של מערכת ההפעלה (SYSINIT) הנמצאת בקובץ IBMBIO.COM שנטען על ידה לזיכרון.

תכנית SYSINIT מאתחלת את מערכת הדיסקים, טוענת את תכניות ההפעלה של התקני החומרה השונים (Device drivers) על פי הפקודות הרשומות בקובץ CONFIG.SYS. אם קובץ זה נמצא בדיסק/דיסקט BOOT, הוא מכיל פקודות להגדרת תצורת החומרה של המערכת (צורת טיפול במסכים, כונני דיסקטים, דיסקים ועוד) והגדרת הפרמטרים לשימוש סביבת מערכת ההפעלה. פירוט של הפקודות בקובץ CONFIG.SYS, מטרתן והדרכים לשפר באמצעותן את התפקוד של מערכת המחשב ניתן למצוא בפרקים המטפלים בהתקנים השונים.

תכנית SYSINIT טוענת גם את קובץ IBMDOS.COM אשר כולל תכנית העורכת את הטבלאות השונות שבשימוש מערכת ההפעלה ומאתחלת את כל ווקטורי הפסיקות



שלב שני בתהליך BOOT

השייכים ל-DOS באיזור RAM המוקצב לכך. בגמר הכנת סביבת העבודה של מערכת ההפעלה מחזירה תכנית העריכה שליטה לתכנית SYSINIT שקראה לה.

תכנית SYSINIT טוענת את מעבד הפקודות COMMAND.COM ומעבירה אליו שליטה. מעבד הפקודות טוען לזיכרון את התכניות השייכות לגרעין מערכת ההפעלה, מפעיל את קובץ AUTOEXEC.BAT אם הוא קיים במערכת הקבצים, או מבקש עדכון תאריך ושעה בלבד, אם לא קיים קובץ כזה. בגמר התהליך נשאר מעבד הפקודות בזיכרון, מציג את הסימן המנחה של מערכת ההפעלה ומוכן לקשר אליה את המשתמש לצורך ביצוע הפעולות שהוא מבקש לבצע.

פירוט של הפקודות החיצוניות והפנימיות של מערכת ההפעלה ניתן למצוא בנספח תקציר פקודות מערכת ההפעלה DOS.

3.7 קובצי משתמש המשתתפים בתהליך BOOT של מערכת ההפעלה

במערכת ההפעלה DOS קיימים שני קובצי הוראות מיוחדים שהמשתמש עורך לצורך תהליך BOOT. מערכת ההפעלה קוראת קבצים אלה בתהליך BOOT ומבצעת את ההוראות הרשומות בהם להגדרת סביבת העבודה של המחשב:

- קובץ CONFIG.SYS מבוצע על ידי תכנית SYSINIT לאתחול מערכת ההפעלה.
- קובץ AUTOEXEC.BAT מבוצע על ידי תכנית COMMAND.COM.

לכל אחד מהקבצים CONFIG.SYS ו-AUTOEXEC.BAT תפקיד מוגדר. המשתמש מגדיר בהם את אופן הפעולה של מערכת המחשב ואת הפרמטרים השונים המגדירים את סביבת הפעולה של מערכת ההפעלה. אם המשתמש אינו מגדיר קבצים אלה, אין הדבר גורם לשגיאת מערכת בתהליך BOOT. התוצאה תהיה שתצורת המחשב וסביבת הפעולה של מערכת ההפעלה יהיו בהתאם לברירת המחדל של מערכת ההפעלה שנקבעה על ידי היצרן.

3.7.1 קובץ CONFIG.SYS

ההוראות שבקובץ CONFIG.SYS מאפשרות למשתמש לקבוע את תצורת מערכת החומרה והתוכנה המבקרת את התקני החומרה. הפקודות נכתבות על ידי המשתמש וניתן לראותן ולהציגן על גבי המסך, להבדיל מקובצי הפעלה בשפת מכונה, שתוכנם אינו מובן למי שאינו בקיא בתכנות.

הפקודות הכתובות בקובץ CONFIG.SYS מאפשרות למשתמש לשפר ולייעל את ביצועי מערכת המחשב מבלי לשנות את הרכב החומרה. הן עושות זאת על ידי הגדרה מחדש של אופן הפעולה של מערכת ההפעלה. הן מפורטות בפרקים המתארים את הציוד שאליו הן מתייחסות. נציין את המאפיינים שלהן:

א. הגדרת תכניות לטיפול בהתקני חומרה שאינם מוכרים על ידי תכנית ROM BIOS.

ב. הגדרת סביבת מערכת ההפעלה: מקום המוקצה בזיכרון הראשי לשמירת נתוני מערכת ההפעלה, מספר קבצים פתוחים של מערכת ההפעלה, גודל מחסנית, מספר קבצים פתוחים אחרונים ועוד.

ג. אופן פעולה והגדרות של מערכת הדיסקים: מספר כוננים במערכת, תכניות Disk Cache, תכניות Virtual Disk ועוד.

ד. הגדרת מערכת התצוגה במחשב: סוגי מסכים ותכניות לטיפול במסך כמו חיקוי למסך ANSI.SYS ועוד.

ה. הגדרה של המקום שבו נמצא מעבד הפקודות של מערכת ההפעלה: קובץ COMMAND.COM.

תקציר פקודות CONFIG.SYS ניתן למצוא בנספח.

דוגמה לקובץ CONFIG.SYS בסיסי:

```
FILES=30
BUFFERS=30
INSTALL=C:\DOS\FASTOPEN.EXE C:50
DOS = HIGH                    (בגירסה 5 ומעלה)
```

3.7.2 קובץ AUTOEXEC.BAT

קובץ AUTOEXEC.BAT מאפשר למשתמש להכין את סביבת העבודה להפעלת היישומים בצורה המתאימה לדרישותיו. הפקודות בקובץ הינן פקודות של מערכת ההפעלה ופקודות של קובצי אצווה.

הפעולות השכיחות בקובץ AUTOEXEC.BAT הן מעבר בין כוננים, טעינת מעטפת (Shell) מתוחכמת לעבודה, הפעלת עברית, טעינת גופנים (fonts) למסך, שינוי הסימן המנחה של מערכת ההפעלה, הגדרת מסלולי גישה (path) לספריות של קובצי מערכת ועוד.

דוגמה לקובץ AUTOEXEC.BAT בסיסי (הסברים ראה בנספח):

```
PROMPT $P$G
PATH C:\DOS;C:\NORTON
EGAHE
HEBREW
PRINT/D:LPT1/Q:30
APPEND C:\TP
DOSKEY                        (בגירסה 5 ומעלה)
DOSSHELL                      (בגירסה 4 ומעלה)
```

3.8 מערכות הפעלה אחרות

עד כאן דנו במערכת הפעלה אחת, מערכת ההפעלה DOS. רוב המחשבים האישיים פועלים בפיקוח מערכת ההפעלה זו, אך יש לזכור שהמחשב האישי ותוכנת DOS אינם יחידה אחת. את מערכת החומרה במחשב האישי, ניתן להפעיל באמצעות מערכות הפעלה אחרות כמו OS/2, VENIX, CPM וכו'. ישנן תכניות, כמו משחקים, שפועלות מדיסקט באופן ישיר מבלי להזדקק לשירותים של מערכת הפעלה כלשהי.

3.9 האם קיימות מגבלות למערכת ההפעלה DOS?

מערכת ההפעלה DOS פותחה בתחילת שנות השמונים עבור מעבדי 8086 ו-8088. מאז חל שיפור משמעותי במעבדים. הוכנסו מעבדי 286, 386, 486 וכבר התבשרנו על מעבדי 586. מערכת ההפעלה DOS שופרה ועודכנה בגירסאות חדשות, החל בגירסה 1 שהיתה ראשונה ועד גירסה 5. מספר עקרונות במערכת ההפעלה DOS נשמרו לאורך כל הגירסאות, אך הם מגבילים אותה ומונעים בעדה מלנצל בצורה מלאה את הטכנולוגיה המתפתחת. בצד ניצול חלקי של המחשבים, מערכת ההפעלה DOS מספקת יתרונות של פשטות ומהירות ביצוע.

מערכת ההפעלה DOS פועלת **באופן פעולה אמיתי**, הקרוי **Real Mode**, או מצב אמיתי, שהינו היחיד הקיים במעבדי 8086 ו-8088. זהו גם אופן הפעולה הפשוט ביותר והמהיר ביותר במעבדי 286 ומעלה. מכאן גם נובעות רוב הבעיות.

המונח **מחסום 640KByte** של הזיכרון הוא למעשה מחסום של 1MByte (640K זיכרון תוכנה + 384 זיכרון חומרה). מחסום 640KByte מתבטא בכך שמערכת ההפעלה DOS אינה מאפשרת **גישה ישירה** לזיכרון מעבר ל-640KByte הראשונים. קיימות מספר דרכים לעקוף את המחסום הזה (ראה בפרק 2 את הדיון על זיכרון הרחבה), אך פתרונות אלה אינם תקניים ואינם נתמכים בצורה מלאה על ידי מערכת ההפעלה. במושג "פתרון לא תקני" אנו מתכוונים לכך שאפשר שהוא לא יפעל בגירסה מתקדמת יותר של מערכת ההפעלה.

הפעלת המחשב האישי בנוהל של **ריבוי משימות** (Multitasking) אינה מהווה מחסום חד משמעי. למעשה, במערכת ההפעלה DOS קיימת אפשרות לכלול פרוצדורות מיוחדות כדי לאפשר ריבוי משימות בגירסאות עתידיות. הבעיה היא שרוב הבעיות הקשות שמערכת הפעלה רבת משימות צריכה לפתור, אפשר לפתור בחומרה **באופן פעולה מוגן** (**Protected Mode**) של מעבדי 286, או באופני פעולה מתקדמים יותר במעבדי 386 ומעלה. פתרון ויישום מערכת רבת משימות בתוכנה איננו כלכלי, כי ביצועי מערכת כזאת נופלים בהרבה ממערכת הפותרת בעיות אלו בחומרה. מערכות כמו DeskView, הן תכניות מעטפת המאפשרות ריבוי משימות תחת מערכת ההפעלה DOS.

3.10 מערכת OS/2 ‏(Operating System/2)

מערכת ההפעלה OS/2 הוכרזה על ידי יבמ בעת ההכרזה על מערכות המחשבים האישיים מסידרת PS/2. מערכת הפעלה זו פותחה כדי לאפשר ניצול מלא של חומרת המעבדים המתקדמים מסוג 286 ומעלה בלבד. השיקול אם להחליף את מערכת הפעלה אינו פשוט כלל ופעמים רבות ההחלפה גם אינה כדאית.

3.10.1 דרישות חומרה במערכת ההפעלה OS/2

מערכת ההפעלה OS/2 יכולה לפעול על מעבדי 80286 ומעלה בלבד. היא משתמשת בזיכרון מינימלי בגודל 1.5MByte עבור מערכת ההפעלה ומחייבת זיכרון נוסף של כ-2MByte לפעולה רגילה. כמות זיכרון זאת אינה לוקחת בחשבון את הזיכרון הנדרש על ידי תכניות המשתמש. מכאן שהמינימום הנדרש לפעולה תקינה של מערכת ההפעלה הוא זיכרון בגודל 3MByte לפחות, ובפועל רצוי זיכרון בגודל 4MByte ומעלה.

מערכת OS/2 **אינה יכולה** לפעול על:
- מחשבי PC/XT ותואמיהם.
- מחשבי PS/2 דגמים 25 ו-30 (לא כולל דגם 30-286).

מערכת OS/2 **תפעל** על מחשבים אלה:
- AT עם מעבד 80286.
- XT286 - דגם בעל ערוץ 8 סיביות עם מעבד 80286.
- מחשבי 386, 386SX ו-486 תואמי PC הקיימים בשוק.
- כל מערכות PS/2 הכוללות מעבדי 286 ומעלה.

לכל הדגמים המוזכרים חייב להיות זיכרון של לפחות 1.5MByte על הלוח הראשי והמצב הרצוי: זיכרון של 4MByte ומעלה. מערכת ההפעלה OS/2 צורכת הרבה ממשאבי המחשב. ולכן, למרות שהיא מסוגלת לפעול גם על מעבדי 80286, מומלץ להפעילה במעבדי 80386 ומעלה בלבד. ודאי שלא כדאי להפעיל אותה במערכות AT ישנות הפועלות במהירות 6-8MHz.

3.10.2 תכונות OS/2

מערכת ההפעלה OS/2 מעניקה שירותים רבים אשר דומים לשירותים שמעניקות מערכות הפעלה במיני מחשבים. היא מיועדת למחשבים האישיים עם המעבדים המתקדמים ביותר ולתחנות עבודה מבוססות מעבדי אינטל. מערכת ההפעלה OS/2 מתאימה לסביבת עבודה רבת משימות ולא מומלץ להשתמש בה במקום מערכת ההפעלה DOS, כדי לקבל שירותים דומים.

א. מערכת OS/2 פועלת ומפעילה יישומים **באופן מוגן** (Protected Mode) של המעבד, אבל היא יכולה לפעול גם **באופן אמיתי** (Real Mode), בעיקר לצורך תמיכה ב-DOS.

ב. המערכת תומכת ב-DOS. האפשרות להפעיל תכניות DOS דרושה מכיון שרוב התכניות למחשבים אישיים בעולם נכתבו לעבודה תחת DOS. אפשר להפעיל תכניות יישומיות שנכתבו עבור מערכת ההפעלה DOS במספר מגבלות:

1) התכנית נכתבה לעבודה תחת גירסה **3.3** DOS ומעלה.
2) התכנית אינה פונה לשגרות של BIOS ומשתמשת רק בפסיקות שירות של DOS.
3) התכנית אינה פונה באופן ישיר להתקני החומרה.
4) התכנית תפעל לאט יותר מאשר היא יכולה לפעול ישירות תחת מערכת ההפעלה DOS.
5) בזמן שתכנית היישום המופעלת תחת DOS בפיקוח OS/2 מועברת לרקע (Background), היא "מוקפאת" ואינה מקבלת זמן מעבד.

ג. מרחב זיכרון בפועל שאליו יכולה התכנית לפנות בגישה ישירה הוא עד 16MByte.

ד. אפשרות לזיכרון בפועל (Virtual Memory). אופן פעולה זה הינו הדמיה של חלק מהדיסק כזיכרון ראשי של המחשב על חשבון מהירות פעולה. מערכת ההפעלה מציגה זיכרון ראשי גדול בהרבה מזיכרון RAM העומד לרשותה בפועל.

ה. מערכת OS/2 תומכת בסביבה רבת יישומים בשיטת ריבוב תכניות (Multiprogramming). היא מאפשרת להריץ ולבצע מספר תכניות יישום במקביל.

ו. מערכת OS/2 תומכת ביישומים רבי משימות (Multitasking). כלומר, באפשרות שתכנית משתמש אחת מבצעת מספר משימות בו זמנית (במקביל). לדוגמה, התכנית תקבל קלט מהמקלדת ומערוץ תקשורת ותכתוב את הקלט גם למסך וגם לדיסק במקביל.

ז. למערכת OS/2 שגרות שירות המנהלות קלט/פלט כמו מדפסות, מסכים עכבר, מקלדת. במהדורה המורחבת EE של OS/2 קיימים שירותים נוספים:

1) **מנהל תצוגה** (Presentation Manager) התומך בחלונות המאפשרים לראות על גבי המסך מספר יישומים הרצים בו זמנית.
2) **מנהל בסיס נתונים** (Database Manager) מספק שירותי בסיס נתונים עם ממשק SQL לתוכנות הרצות במערכת.
3) **מנהל התקשורת** (Communication Manager) תומך בשירותי התקשורת הבאים:
   - רשתות LAN (לא כל סוגי הרשתות).
   - תקשורת טורית סינכרונית/אסינכרונית.
   - חיבורים ייעודים למחשבים מרכזיים (Main Frame).

ח. מערכת OS/2 מספקת מיגוון של אפשרויות תקשורת פנימיות במערכת:
- תקשורת בין משימות שונות של אותה תכנית יישום.
- תקשורת בין תכניות יישום שונות הרצות בו זמנית במערכת.
- תקשורת בין משימות המשתמש לבין משימות של מערכת ההפעלה.

ט. מערכת OS/2 מספקת הגנה על הזיכרון. מערכת OS/2 מגינה על עצמה וגם על תכניות המשתמש ואינה מאפשרת לתכניות השונות המתבצעות במקביל "להפריע" זו לזו. תוכנת יישום הפועלת תחת מערכת הפעלה DOS אינה מגינה על עצמה ועל הזיכרון. כלומר, מערכת DOS מתירה לתכניות השונות לבצע כל פעולה שהיא על מערכת החומרה במחשב ובכך כל תכנית משתמש יכולה לפגוע בה.

3.10.3 דרישות תוכנה במערכת OS/2

למערכת OS/2 ישנם מהדרים (Compilers) המיועדים לתכניות בשפות התכנות השונות. בדרך זו היא פוטרת את המשתמש מטיפול בנושאי התפעול המיוחדים לתכנית תחת OS/2. כדי לאפשר לתכנית לנצל באופן מלא את תכונות OS/2 ולקשור אותה לתכניות אחרות הפועלות במערכת, יש להכיר את מערכת ההפעלה בצורה מעמיקה יותר וללמוד את הקשרים בין מרכיביה כדי לנצל את אפשרויותיה הרבות.

תקשורת בין תהליכים ויישומים שונים במערכת OS/2 אפשריים באמצעות אתתים (Semaphores), אותות (Signals), צינורות (Pipes), תורים (Queues) ובאמצעות שיתוף דינמי בזיכרון (Shared Memory).

כדי להסב תכנית שפועלת במערכת ההפעלה DOS לפעולה תחת OS/2 יש לבצע מספר דברים:

1. להמיר את כל הפקודות הישירות להפעלת התקני חומרה (אם יש כאלו) לקריאות באמצעות שירותי מערכת ההפעלה.
2. את קריאות השירות לשגרות BIOS ולשגרות של DOS יש להמיר לקריאות התקשרות דינמיות של הליכי הקלט/פלט המתאימים. כדי לבצע את ההמרה יש לפעול לפי ההנחיות בספר OS/2 Programmers Reference.

3.10.4 השוואה בין OS/2 לבין DOS

במה דומות המערכות?

א. מערכת ניהול הקבצים. כל קובץ שנכתב במערכת DOS יכול להיקרא על ידי OS/2 ולהיפך.

ב. בשתי המערכות קיים קובץ CONFIG.SYS המגדיר את תצורת המערכת. ב-OS/2 ישנן אפשרויות נוספות להגדרת תצורת המערכת המורחבת וחלוקת המשאבים הקיימים בה.

ג. בשתי המערכות קיימת אפשרות לבצע תכנית אצווה בסוף תהליך BOOT באמצעות קובץ דוגמת AUTOEXEC. במערכת OS/2 שם הקובץ הוא STARTUP.CMD. סיומות קובצי אצווה ב-OS/2 הם CMD ולא BAT.

דוגמה לקובץ STARTUP.CMD:

התכנית	הסברים
prompt PG	כמו ב-DOS
path c:\;c:os2;c:\mydir	כמו ב-DOS
start MyBatch	הרץ קובץ אצווה בשם MyBatch.cmd
myprog	הרץ תכנית myprog במקביל לפעולה הקודמת
editor	הרץ תכנית עריכה במקביל לפעולות קודמות
exit	צא. התכניות שהופעלו קודם לא יופסקו.

ד. תהליך BOOT דומה מאוד בשתי המערכות ומוכתב בעיקרו על ידי רכיבי החומרה של ROM BIOS. בסיום התהליך ב-OS/2 המעבד יהיה במצב מוגן.

ה. במערכת OS/2 יש אפשרות להפעיל מחיצה דמויית DOS ומכאן שמערכת ההפעלה DOS כלולה במערכת OS/2. יש לזכור שהדמיית DOS של OS/2 אינה מלאה ומחייבת שינויים מסוימים בתכניות.

במה הן שונות?

א. מערכת OS/2 פועלת באופן מוגן ומנצלת את כל האפשרויות של אופן עבודה זה. מערכת DOS פועלת באופן אמיתי ולכן היא "זריזה" ופשוטה יותר. מערכת OS/2 מחייבת החלטות רבות, בדיקות, מעבר בין תכניות, ניצול משאבים וחלוקת משאבים ולכן היא איטית יותר ממערכת DOS.

ב. פניות של תכנית יישום למערכת ההפעלה.
במערכת DOS כל הפניות לקבלת שירות ממערכת ההפעלה מבוצעות על ידי פסיקות שירות, ובעיקר פסיקת השירות 21Hex.
במערכת OS/2 הבקשות ממערכת ההפעלה מבוצעות על ידי קריאה לשגרות של תהליכים במערכת ההפעלה, מכיון שיש קשרים בין התהליכים במערכת. לדוגמה:

כל הקריאות לטיפול בלוח המקשים מתחילות באותיות KBD, כמו
קריאת תו מלוח מקשים KBDCHARIN
כל הקריאות לטיפול במסך מתחילות באותיות VIO, כמו
כתיבת מחרוזת תווים למסך VIOWRTCHARSTR
כל הקריאות לטיפול בקבצים מתחילות באותיות DOS, כמו
פתיחת קובץ DOSOPEN
כל הקריאות לטיפול בעכבר מתחילות באותיות MOU, כמו
הצגת העכבר על המסך MOUDRAWPTR

ג. יש שוני באופן הפעולה של תכנית ההפעלה להתקנים (Device Driver). תכנית ההפעלה במערכת OS/2 מקבלת בקשה לקלט/פלט בעזרת שגרת האסטרטגיה של ההתקן(ב-DOS שגרת האסטרטגיה ריקה תמיד). שגרה זו רושמת את הבקשה וכאשר ההתקן **מתפנה**, מתבצעת פעולת קלט/פלט על ידי שגרת הפסיקה המנוהלת באמצעות מנגנון הפסיקות. ניהול התקנים באמצעות מנגנון פסיקות מאפשר למערכת OS/2 לנצל את משאבי המחשב

בצורה אופטימלית, מכיון שהמעבד ממשיך לטפל בתכניות אחרות ואינו צריך להמתין לסיום פעולת ההתקן.

ד. ניהול הזיכרון.
במערכת DOS ניהול הזיכרון פשוט ומתאים לזיכרון קטן. החלק התחתון של הזיכרון מכיל את טבלת הפסיקות, אחריה נמצאת מערכת ההפעלה, תכניות תושבות (TSR) ולבסוף נמצאת תוכנת המשתמש.

במערכת OS/2 הזיכרון מנוהל בצורה דינמית. הסגמנטים שמוקצים לכל תהליך מוקצים בצורה דינמית ומקושרים לתהליכים אחרים. הקצאת הזיכרון, הקישור בין תהליכים, הגנה והרשאה לתהליכים מנוהלים אך ורק באמצעות מערכת ההפעלה.

ה. קישור בין תהליכים.
במערכת DOS אין דרך מוכתבת לקישור בין התכניות השונות הנמצאות בזיכרון. הקישור היחידי הניתן הוא בעזרת פסיקות תוכנה. לעומת זאת, במערכת OS/2 הוקדשה מחשבה רבה (כמו בכל מערכת הפעלה מסוג Multitasking) לקישור בין תהליכים. כל פעילות במחשב היא תהליך ומערכת ההפעלה מנתבת את הפעילות בין התהליכים השונים.

3.10.5 סיכום

מערכת ההפעלה OS/2 מספקת סביבת עבודה הדומה לסביבת עבודה במיני מחשבים. אנשי מחשבים מרקע של מחשבים גדולים ימצאו הרבה קווים משותפים לסביבת עבודה זו. לעומתה, מערכת DOS פשוטה, קטנה וזריזה. מערכת DOS מיועדת להפעיל את המחשב עבור משתמש בודד עם תוכנית אחת ותמיכה מינימלית בסביבת העבודה.

יש לזכור שמערכת הפעלה אינה מטרה, אלא אמצעי להפעלת תכניות ישום. מכאן, שמערכת הפעלה אלגנטית, משוכללת ובעלת יכולת גבוהה, אינה צריכה להיות שיקול מכריע למעבר למערכת הפעלה חדשה. השיקול המכריע הוא אופי היישומים שאנו רוצים להפעיל באמצעות המחשב האישי ותחת איזה מערכת הפעלה הם מסוגלים לפעול ביעילות ולתת לנו את השירות המירבי.

כיום, מרבית היישומים הקיימים פועלים תחת מערכת DOS ורוב היישומים שמפתחים הם עבור מערכת זו. כלומר, על אף הציפיות שמערכת OS/2 תסחוף אליה את מרכז הכובד של פיתוח התוכנה, הדבר עדיין לפנינו.

מוצר חשוב אשר הולך וכובש את מקומו בשוק הוא **WINDOWS**. מוצר זה יוצר עבור המשתמש סביבת עבודה הפועלת מעל למערכת ההפעלה DOS ונותנת לו שירותים אשר מתחרים בחלק מהשירותים של OS/2. WINDOWS נוגסת בשוק של OS/2 ותפוצתה רבה מאוד.

WINDOWS יוצרת סביבת חלונות גרפיים המנוהלים גם באמצעות העכבר וגם באמצעות המקלדת. ב"תוך" כל חלון כזה אפשר להריץ תוכניות יישום נפרדת ובכך, הופכים את המחשב למערכת של ריבוי משימות (Multitasking). הפעלת החלונות הדינמית הנעשית בצורה סטנדרטית בתוך תוכנית היישום, הופכת את הלימוד והשימוש בטכניקה זו לפשוטים ואחידים. WINDOWS הופכת כיום למוצר הדגל של חברת מיקרוסופט.

פרק 4

הממשק בין המשתמש לבין המחשב האישי

המחשב האישי, או ליתר דיוק - המעבד שלו, אינו מערכת מחשב. כלומר, על מנת שניתן יהיה להפעיל את המחשב האישי כמערכת שלמה ולאפשר למשתמש להתקשר ולנהל דו-שיח, יש צורך בממשק (Interface) כדי לקיים קשר זה. הממשק יאפשר למשתמש להכניס נתונים ופקודות באופן נוח, או לראות באופן ברור ונעים לעין פלט והודעות המשמשות לבקרת פעולתה של מערכת המחשב. הממשק גם יאפשר למשתמש להוציא את המידע מהמחשב על גבי נייר לשימוש חיצוני. ניתן לומר, שללא ממשק מתאים לא ניתן להפעיל את מערכת המחשב.

ככל שהממשק המקשר בין המשתמש למחשב האישי יהיה נוח יותר ופשוט יותר, כך תהיה הפעולה נעימה וקלה יותר. ממשקי הקשר מגוונים ומכילים קשת שלמה ורחבה של מוצרים והתקנים מכל הסוגים, שנועדו לאפשר למשתמש עבודה נוחה ויעילה מול המחשב האישי. מטרת פרק זה להסביר את סוגי ההתקנים העיקריים הניתנים לחיבור כממשק בין המשתמש למחשב ואת אופן הפעולה שלהם. בנוסף נסביר כיצד מתקשרים המשתמש והמתכנת להתקנים אלה ומבקרים את אופן הפעולה שלהם, הן דרך פקודות מערכת ההפעלה DOS והן דרך פקודות תכנות שונות.

בפרק זה נעסוק בהתקנים שמקשרים את המשתמש למחשב:

* המסך וכרטיס התיאום למסך.
* לוח המקשים.
* המדפסת ומתאם המדפסת המקבילי.
* תקשורת טורית אסינכרונית RS232 (העכבר).

התקן התקשורת הטורי מוצג בפרק זה כהתקן קישור לעכבר. בנוסף לשימושיו הבסיסיים כממשק קלט, קיימים עבורו שימושים נוספים שבהם נדון בהרחבה בפרק 7.

4.1 מסכים ומתאמי מסך

המסך במערכת המחשב האישי משמש אמצעי פלט, אשר בעזרתו רושמות מערכת ההפעלה ותכניות היישום השונות הוראות למשתמש, נתונים שונים, הודעות שגיאה וכו'. בנוסף, משמש המסך כאמצעי בקרה למשתמש. כל הנתונים שמוקשים כקלט מופיעים על גבי המסך ומשמשים כבקרה חזותית לנכונותם, לפני הזנתם למחשב. באופן זה מקבל המשתמש חיווי מיידי על שגיאות שביצע ויכול לפעול בצורה של דו-שיח עם מערכת ההפעלה ותכניות היישום השונות.

באמצעות ראיה של הנעשה על גבי המסך יכול המשתמש להגיב ולבצע שינויים בנתונים, לקבל החלטות חדשות ולהיות מעורב בנעשה. וכל זאת, תוך כדי התהוות המידע במהלך העיבוד ולא לאחר פרק זמן, שבו התגובה עלולה להיות מאוחרת ולא מתאימה.

תהליך זה, הנראה פשוט ובסיסי לכאורה, לא תמיד היה כך. בתחילת דרכו של המחשב לא היו בידי המשתמש כלים לפעולה אינטראקטיבית והוא היה נאלץ לחכות פרק זמן ארוך של שעות/ימים עד שקיבל תגובה לבקשותיו. הדבר לא היה לפני עשרות שנים, אלא בשנות ה-70, לפני כ-20 שנים.

המסך קשור למחשב האישי באמצעות מתאם מסך, כרטיס המתחבר לאחד ממחברי ההרחבה הנמצאים על הלוח הראשי (לא חשוב לאיזה מהם). במספר מחשבים אישיים כדוגמת PS/1, או PS/2 של יבמ ומחשבים תואמים מתקדמים, לא מופיע מתאם המסך ככרטיס נפרד, אלא כחלק מהחומרה הקיימת על גבי הלוח הראשי של המחשב.

המסכים השונים הקיימים כיום מסוגלים להציג מידע טקסטואלי (אותיות ותווים) ומידע גרפי. ההבדלים בין מגוון המסכים הקיימים כיום באים לידי ביטוי באיכות ההצגה הטקסטואלית, במספר הצבעים שניתן להציג, באיכות הצגת המידע הגרפי ועוד.

4.1.1 המסך במחשב האישי

התכונה העיקרית של מסך המחשב (Screen, או מרקע) היא כמות המידע המירבית המופיעה על פניו. המסך מחולק כמטריצה דו-מימדית של נקודות (Pixels), אשר מספרן (גודל המטריצה) קובע את הרזולוציה של המסך. לדוגמה, מסך מונוכרום מחולק למטריצה שמכילה 720 נקודות אופקיות (לרוחב) ו-350 נקודות אנכיות (לגובה). מכאן, שהרזולוציה של מסך זה היא 720x350.

כדי לקבל את כמות המידע המוצג על גבי המסך יש לכפול את הרזולוציה בכמות הגוונים/צבעים שניתן להציג על גבי המסך בכל נקודה ונקודה של המטריצה. לדוגמה, במסך מונוכרום ניתן להציג בכל נקודה של המטריצה שני צבעים שונים בלבד, ומכאן שכמות המידע במסך מונוכרום היא 720x350x2 (נקודות, או סיביות).

במסך EGA הרזולוציה היא 640x350. רזולוציה זו נמוכה מזו של מסך מונוכרום, אך כל נקודה במסך EGA יכולה לקבל 16 צבעים שונים ומכאן שכמות המידע בו היא 640x350x16. כמות מידע זו גדולה בהרבה מכמות המידע במסך מונוכרום. ככל שכמות המידע שניתן להציג על גבי המסך גדולה יותר, מחירו של המסך גבוה יותר.

המסכים נחלקים לשתי קבוצות עיקריות:

* מסכי גביש נוזלי

אלה הם מסכים שטוחים המיועדים למחשבים ניידים/נישאים. מסכים אלה יקרים יותר ופחות נוח לצפות בהם. הם בנויים העיקרון של המסך בשעונים דיגיטליים. מסך הגביש הנוזלי מחולק למטריצה של גבישים

נוזליים. הגביש הנוזלי נעשה שקוף כאשר מחברים אותו למתח חשמלי ואז מופיע עליו צבע הרקע (בדרך כלל לבן). גביש נוזלי אשר אינו מחובר למתח חשמלי אינו שקוף, ולכן הוא משאיר נקודה שחורה על גבי המסך.

על ידי חיבור נקודות מסוימות מהמסך למתח חשמלי וניתוק הנקודות האחרות, נוצרים על גבי המסך איזורים שקופים המחוברים למתח ואיזורים כהים, אשר אינם מחוברים למתח. כך ניתן להציג תמונות וצורות שונות על גבי המסך (אותיות, גרפיקה ועוד). מסכים אלה מחייבים כרטיס מתאם מיוחד.

* מסכי שפופרת קרן קטודית (CRT)

מסכי שפופרת קרן קטודית הם המסכים הרגילים הנפוצים בשוק. מסך קרן קטודה (CRT) מונוכרום פועל כדוגמת מסך הטלוויזיה הביתי. קרן אלקטרונית הנשלחת מהחלק האחורי של המסך מכוונת אל החלק הקדמי וסורקת אותו בשורות אופקיות, שורה אחר שורה מלמעלה למטה. הפקודות הנשלחות מהכרטיס המתאם אל המסך מציינות לקרן אם יש צורך להאיר את הנקודה אותה היא סורקת על פני המרקע.

כדי ליצור תיאום (סינכרון) בין הכרטיס של המחשב לבין הקרן, שולח הכרטיס המתאם שתי אותות תיאום: אות הסינכרון האופקי (Horizontal) מסמן לקרן התחלת סריקה של שורה אופקית חדשה. אות הסינכרון האנכי (Vertical) מסמן תחילת סריקה של המסך מנקודה שמאלית עליונה.

אות הסינכרון האנכי יופיע אחת לכל מספר אותות סינכרון אופקיים, על פי מספר שורות המגדירות את הרזולוציה האנכית במסך. אות הסינכרון האנכי חשוב במיוחד ליציבות התמונה ולכן התדר שלו גבוה יותר. הבהוב של התמונה, אשר קיים בדרך כלל במסך CGA, נגרם בשל תדר סריקה נמוך. ההבהוב גורם לעייפות העיניים אצל אלה אשר עובדים זמן רב מול המסך.

השאלה המתבקשת היא: מדוע לא נעלה את תדר הסריקה ונשפר את יציבות התמונה? התשובה נעוצה בעובדה שהעלאת תדר הסריקה תגרום לקרן לשהות זמן קצר יותר על כל נקודה. כדי שנוכל לראות את המידע הנשלח אל אותה נקודה על גבי המסך ולשמור על הפרדה בין הנקודות הסמוכות, יש להשתמש במסכים בעלי יכולת פעולה מהירה יותר (בעלי רוחב פס, Band Width, גדול יותר), כלומר מסכים יקרים יותר.

מסכי קרן קטודה צבעוניים פועלים באופן דומה למסכי קרן קטודה מונוכרום. במסכי קרן קטודה צבעוניים יש שלוש קרניים אלקטרוניות לעומת קרן אחת בלבד במסכי קרן קטודה מונוכרום. קרן אחת מאירה בצבע אדום, אחת מאירה בצבע ירוק ואחת מאירה בצבע כחול (מסכי Red - RGB, Green, Blue). שילוב מתאים בעוצמת ההארה של כל קרן יכול ליצור את כל גווני הצבעים הקיימים בקשת הצבעים, כולל הצבעים שחור ולבן.

במסכי מונוכרום וצבע רגילים (עד רמת CGA) הכבל בין הכרטיס למסך מעביר אותות סינכרון וגם אותות נתונים בכל אחד מ-4 החוטים שלו (Intensity, Red, Green, Blue). כך מתקבלות רמות של 0 ו-5 וולט בלבד ולא רמות ביניים כלשהן. שתי רמות מתח אלו מגבילות את כמות הצבעים שניתן להציג. ואמנם, באמצעות 4 חוטים המתחברים בין המתאם

למסך ניתן להגדיר במסך CGA עד 16 צבעים שונים (מספר הצירופים של 0 ו-1 בארבעה חוטים הינו 16).

מסכי RGB מאפשרים חיבור של רמות מתח כלשהן בתחום שבין 0 ל-5 וולט ועל כן הם מסוגלים להציג רמות וגווני צבע רבים יותר ממסכים רגילים. מסכי RGB הינם בעלי רוחב פס של 40-60MHz ומתאימים לכרטיסי מסך EGA, VGA ומעלה. מחיר של מסך RGB גבוה בהרבה ממחיר מסך רגיל.

המסכים מקבלים ממתאם המסך את כל המידע הנדרש להצגה חזותית של התווים או הגרפיקה באמצעות כבל המכיל מספר חוטים. המסכים אינם זוכרים או מאחסנים את המידע, אלא מציגים אותו בלבד. המידע נראה מוצג באופן קבוע על גבי המסך, מכיון שהכרטיס המתאם שולח את המידע למסך מספר פעמים בשנייה על פי תדר הסריקה האנכי.

מספר פין	CGA מחבר 9 פין	MONO מחבר 9 פין	EGA מחבר 9 פין	SVGA מחבר 15 פין
1	GROUND	GROUND	GROUND	RED
2	GROUND	GROUND	SECONDARY RED	GREEN
3	RED		PRIMARY RED	BLUE
4	GREEN		PRIMARY GREEN	GROUND
5	BLUE		PRIMARY BLUE	GROUND
6	INTENSITY	INTENSITY	INTENSITY / GREEN	RED GROUND
7		VIDEO	BLUE / MONO VIDEO	GREEN GROUND
8	HORIZONTAL	HORIZONTAL	HORIZONTAL	BLUE GROUND
9	VERTICAL	VERTICAL	VERTICAL	
10				GROUND
11				SYNC. GROUND
12				
13				HORISONTAL SYNC.
14				VERTICAL SYNC.
15				

פירוט חיבורים בפתיל מסך

4.1.2 כרטיס תיאום למסך (מתאם מסך)

כרטיס התיאום למסך (או מתאם המסך) מתחבר למחשב באמצעות אחד ממחברי ההרחבה שעל הלוח הראשי במחשב. ישנם סוגים שונים של כרטיסי תיאום למסך ולכן, בתהליך BOOT יש לציין למערכת ההפעלה איזה סוג של **מתאם מסך** (לא סוג מסך) מותקן במערכת. אם מותקנים מספר מתאמי מסך, יש לציין למערכת ההפעלה איזה מהם ישמש כברירת מחדל להצגת מידע של מערכת המחשב (הודעות שגיאה, דו-שיח בתהליך BOOT, דו-שיח עם מערכת ההפעלה וכו').

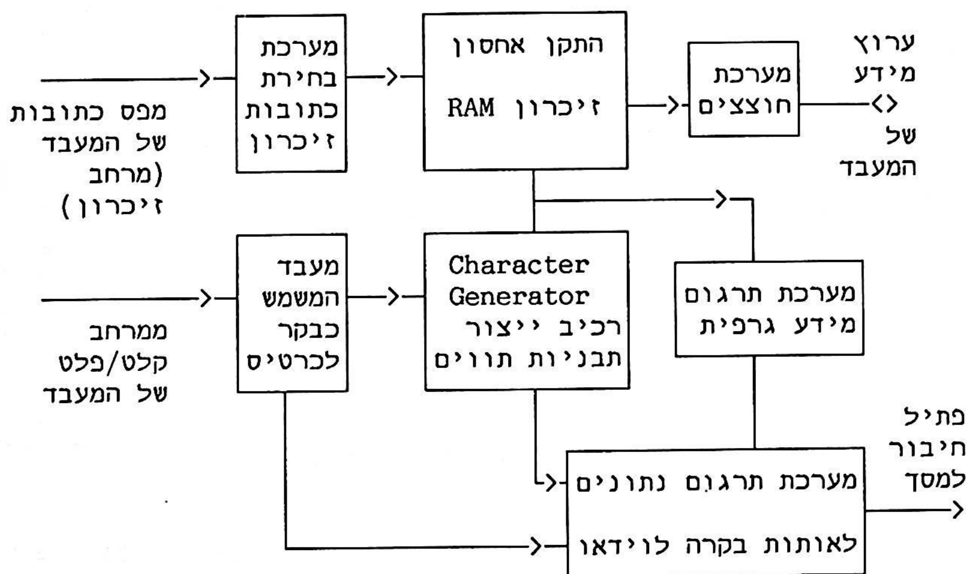
במחשבים המבוססים על מעבדי 8088/8086 ניתן לציין את סוג מתאם המסך על ידי כיוון מתגים על גבי הלוח הראשי (DIP Switches, פרק 2). במחשבי AT ומעלה ניתן לציין את סוג המסך במתגים, או על ידי הגדרה בתהליך Setup (ברכיב השעון CMOS RAM). הצורך בהגדרת סוג מתאם המסך נובע מכמה סיבות:

* מתאמי המסך השונים נבדלים זה מזה בחומרה המרכיבה אותם. הם ממופים במרחב הזיכרון של המעבד בכתובות שונות במרחב של 1MByte של המעבד.

לכן, גישה לכתובת של מתאם מסך שלא קיים במערכת המחשב, לא תאפשר הצגת נתונים על גבי המסך הקשור אליו.

* קיים הבדל בפקודות הבקרה והשליטה על מתאמי המסך השונים. פקודות תוכנה שונות המאפשרות לבצע פעולות זהות יוצרות חוסר תאימות מסוים וגם קושי רב. הן מחייבות את מערכת ההפעלה ותכניות היישום השונות להכיר את סוג מתאם המסך הנמצא במחשב. רק כך הן תוכלנה להפעיל את מתאם המסך בצורה נכונה ולהציג את הנתונים על גבי המסך באופן הרצוי.

* מתאמי המסך נבדלים זה מזה בממשק (Interface) החומרה. יש מתאמי מסך המתחברים למחשב בממשק ברוחב 8 סיביות ויש מתאמים המתחברים למחשב בממשק של 16 סיביות. במחשבים עם ערוץ נתונים של 16 סיביות ניתן לחבר את שני סוגי המתאמים. רוחב הנתונים הנכתבים למסך הוא בדרך כלל 16 סיביות, ולכן בממשק ברוחב 16 סיביות זמן הכתיבה למסך קצר יותר מאשר בממשק של 8 סיביות.



מבנה בסיסי של מתאם מסך

4.1.2.1 פירוט המרכיבים השונים

* מערכת בחירת כתובות זיכרון

זוהי מערכת חומרה הממפה את הכרטיס המתאם למסך במרחב הזיכרון של התקני החומרה של המעבד, על פי המקום שיועד לו על ידי יצרני החומרה/תוכנה. מערכת בחירת הכתובות מאפשרת העברת מידע מהכרטיס למעבד ולהיפך, רק כאשר המעבד שולח כתובת על ערוץ הכתובות השייכת למרחב הכתובות שבו ממופה הכרטיס.

* זיכרון RAM

גודל זיכרון RAM של הכרטיס תלוי בסוג מתאם המסך. המעבד יכול לכתוב/לקרוא נתונים לזיכרון זה, מכיון שהזיכרון ממופה במרחב הכתובות של התקני החומרה. בזיכרון RAM מאחסן המחשב את הנתונים שיש להציג על גבי המסך (נתונים גרפיים או תמליל). המעבד ניגש לזיכרון באמצעות פקודות גישה, בצורה ישירה ובמהירות גבוהה כדי לכתוב/לקרוא נתונים. בזיכרון זה אין בקרת שגיאות (Parity).

מרחב זיכרון RAM במחשב משמש גם את מתאם המסך. מתאם המסך לוקח מתוך הזיכרון את הנתונים שיש להציג (נתונים גרפיים או תמליל) ומתרגם אותם למידע הנשלח להצגה במסך.

יכולים להיות מקרים שבהם מתאם המסך מכיל יותר זיכרון ממה שמוקצה לו במרחב הזיכרון במחשב. מכיון שאסור לתפוס שטח זיכרון גדול ממה שהוקצה, משתמשים בשיטת הדפדוף. בשיטה זו הזיכרון מחולק ל"דפים" (Pages), אשר נטענים ומוחלפים זה אחר זה לפי הצורך. כך, בכל עת יוצג מול המעבד דף (שטח זיכרון ב-RAM) בגודל מירבי המוקצה למתאם המסך. התכנית המאפשרת גישה והפעלה של מתאם המסך נמצאת ב-ROM BIOS (פסיקת שירות 10Hex). למתכנת ניתנת האפשרות לכתוב ישירות למתאם המסך באמצעות תכנית השירות של BIOS, או דרך פסיקות שירות של מערכת ההפעלה DOS.

מערכת ההפעלה משתמשת גם כן בתכנית BIOS לצורך תרגום בקשת השירות של המתכנת להצגה על גבי המסך, אך מספקת ממשק תוכנה פשוט יותר ואפשרויות רחבות יותר למתכנת. ה"מחיר" שהמתכנת משלם עבור ממשק מערכת ההפעלה הוא בזמן ביצוע ארוך יותר של התכנית.

סוג מתאם מסך	כתובת זיכרון (RAM)	מרחב קלט/פלט	גודל RAM מוקצב
מתאם מסך צבעוני CGA	B800:0	3D0-3DF	16 KByte
מתאם מסך ש/ל MONO ללא גרפיקה	B000:0	3B0-3BF	32 KByte בפועל 4K
מתאם מסך צבע EGA	A000:0 גרפיקה B800:0 טקסט		64 KByte התקן כיום 256 KB
מתאם מסך צבע SVGA	A000:0 גרפיקה B800:0 טקסט		512KB עד 1MB

זיכרון RAM שעל גבי מתאם המסך ממופה במרחב הזיכרון של המחשב, ולכן ניתן לגשת ישירות לזיכרון זה כדי לכתוב, או לקרוא ממנו נתונים. מכיון שהנתונים הנמצאים ב-RAM של מתאם המסך מוצגים ישירות על גבי המסך, ניתן לשנות את המידע המוצג על ידי כתיבת נתונים ל-RAM. כתיבה ישירה למתאם המסך מהירה יותר מאשר כתיבה באמצעות BIOS, או

באמצעות מערכת ההפעלה. אך פעולה זו מחייבת את המתכנת להכיר את צורת הכתיבה ב-RAM, את מרחבי הכתובות של מתאמי המסך השונים וצורת הארגון של זיכרון RAM במתאם.

כתיבה ישירה למסך אינה מומלצת בשל קשיים אפשריים של תאימות תוכנה עם פיתוחים עתידיים של החומרה. עם זאת, בתכניות שבהן משך ביצוע התכנית הינו קריטי, הכרחי לכתוב ישירות למתאם המסך.

* מחולל תווים (Character Generator)

המחולל מייצר את הצורות הגרפיות של התווים. לכל תו מ-256 התווים הנתמכים בו-זמנית על ידי הכרטיס המתאם למסך יש צורה מסוימת ומוגדרת מראש, אשר מוגדרת בתבנית מטריצה של W x H נקודות (גודל התבנית נקבע על פי סוג המתאם). הערך W מגדיר את מספר הנקודות לרוחב התבנית (Width) והערך H מגדיר את מספר הנקודות בגובה.

שליחת תבנית נקודות למסך תגרום להצגת התו המתאים על פי הקביעה של מחולל התווים. הוא נעזר לשם כך בזיכרון RAM שבמתאם המסך, המכיל את התווים שיש להציג. על גבי הכרטיס המתאם למסך נמצא רכיב ROM, המכיל את התבניות המתאימות לכל אחד מהתווים. מחולל התווים לוקח מרכיב זה את התבניות הדרושות לו כדי להציג את התווים על גבי המסך.

אוסף התבניות ברכיב ROM נקרא **גופן** (FONT). בכרטיסי CGA ומונוכרום קיים גופן אחד בלבד, וכדי לשנותו (להוסיף תווים בעברית למשל), יש צורך להחליף אותו. פעולת ההחלפה מורכבת ומצריכה ציוד חיצוני, כמו צורב זיכרונות. במרבית המקרים אין למשתמש רגיל יכולת לבצע את פעולת ההחלפה.

בכרטיסי EGA ומעלה קיימת אפשרות לשנות את הגופן המוצג על גבי המסך ולהציג במקומו גופן אחר על פי בחירתנו. הגופן האחר יכול להיות עיצוב אחר של אותם תווים, או אוסף תווים המותאם לדרישות שפה מסוימת. המשתמש, ובודאי כל אדם מיומן, יכול לטעון את הגופן הרצוי לזיכרון RAM של הכרטיס המתאם באמצעות תוכנה.

בכרטיסי EGA ומעלה המשתמש יכול גם לעצב גופן על פי בחירתו והעדפתו האישית. על מנת לאפשר לו לעשות זאת, מספקים מספר יצרנים של מתאמי מסך EGA ומעלה תכנית לעריכה גרפית של גופנים. באופן זה ניתן לטעון גופן כלשהו לזיכרון RAM של מתאם המסך ולהציג אותו במקום הגופן הבסיסי שמסופק על ידי היצרן ברכיב ROM.

כאשר מתאם המסך נמצא במצב של הצגת טקסט, מרחב הטקסט (ראה טבלה למעלה) מוצג על גבי המסך. לכל נתון/תו המוצג על גבי המסך מוקצה במרחב הטקסט של המתאם מלה אחת בזיכרון ברוחב 16 סיביות. לדוגמה, כאשר מתאם המסך פועל באופן פעולה של 80x25 (כלומר 25 שורות בנות 80 תווים בשורה), יוקצו למסך 25x80=2000 מלים שהן 4000 בתים.

לאיזור הזיכרון במתאם המסך יכולים לגשת גם המעבד וגם מערכת החומרה במתאם המסך. כדי להציג את התווים בצורה נכונה על גבי המסך יש צורך לתרגם אותם באמצעות מחולל התווים - Character Generator. התווים

"מתורגמים" לתבניות של נקודות המגדירות את צורת התו. לשם כך דרוש תיאום בין:

- מיקום הקרן במסך.
- מיקום התו בזיכרון המתאם. מיקום זה קובע היכן יירשם התו על גבי המסך.
- זיכרון ROM הקובע את תבנית התו על פי ערך התו שיש להציג וצורת הגופן הקיים ב-ROM.

את השליטה על התיאומים האלה לשם הצגת הטקסט מבצע מחולל התווים.

* בקר המסך

מתאם המסך כולל את **בקר המסך** (CRT Controller). רכיב זה נמצא על כרטיס מתאם המסך ותפקידו לשלוט על פעולת הכרטיס ועל אופני ההצגה השונים שהכרטיס מאפשר. בקר המסך ניתן לתכנות ועל כן אפשר לכתוב פקודות שונות כדי להפעיל אותו באופנים שונים: טקסט, גרפיקה, רזולוציה שונה וכו'.

* מערכת תרגום גרפית

מערכת התרגום הגרפית הינה רכיב חומרה, אשר תפקידו לתרגם את הנתונים הנמצאים בזיכרון RAM של המתאם לנקודות המוצגות על גבי המסך. בהצגה גרפית מתאם המסך מתרגם כל נקודה באמצעות מערכת התרגום הגרפית, ואילו בהצגת טקסט מתאם המסך מתרגם כל תו באמצעות מחולל התווים.

המערכת הגרפית מקצה לכל נקודה מספר סיביות בתוך ה-RAM, בהתאם למספר הצבעים שניתן לצבוע בהם כל נקודה. על כן, כמות הנתונים שיש לכתוב למתאם המסך על מנת להציג מסך גרפי מושלם גדולה בהרבה מכמות הנתונים שיש לכתוב למתאם המסך על מנת להציג מסך טקסט מושלם. בשל כמות הנתונים הקטנה יותר, הצגת טקסט מהירה הרבה יותר מהצגת גרפיקה.

* מערכת חוצצי זיכרון

מערכת זו מאפשרת לכתוב/לקרוא מהכרטיס המתאם בהתאם לבקשת המעבד. מכיון שזיכרון RAM משמש גם את המחשב וגם את הכרטיס, יש צורך במערכת מתאמת, אשר תמנע התנגשות ביניהם ותאפשר למעבד המחשב לקרוא/לכתוב מידע לכרטיס בלא לפגוע בתהליך הצגת הנתונים על גבי המסך.

* מערכת תרגום לאותות בקרה לוידאו

המערכת ממירה את הנתונים הנשלחים מבקר המסך לאותות וידאו השולטים על פעולת הקרן במסך. אותות הוידאו שונים בהתאם לסוגי המסך השונים המחוברים לכרטיס. מתאמי מסך מסוימים יכולים להפיק אותות וידאו למספר סוגים של מסכים באמצעות מערכת תרגום משוכללת. בעת הפעלת המסך יש להורות לכרטיס המתאם את סוג המסך שבו משתמשים.

4.1.2.2 הסמן — Cursor

הסמן הנראה על גבי המסך, מסמן למשתמש היכן יירשמו על המסך הנתונים שיוקלדו. הסמן מופיע רק באופן ההצגה של תווים (טקסט). כל נתון שיוקש מלוח המקשים יירשם במקום שבו הוא נמצא. כאשר הסמן מגיע לסוף השורה, הוא מועבר מידית לתחילת שורת התווים הבאה. פעילות הסמן, מיקומו על המסך וצורתו - מוכתבת על ידי תכנית ROM BIOS המטפלת בגישה למסך.

באמצעות תכניות שירות של BIOS ניתן לקבוע את צורתו של הסמן וגודלו ולהניע אותו על גבי המסך לכל מקום שנרצה. במידה וקיימת אפשרות להציג יותר מדף אחד על המסך (ראה בהמשך, נושא הצגת דפים), אחראית תוכנת BIOS לנהל את הסמנים של כל הדפים שאפשר ליצור ולשמור. צורת הסמן וגודלו יהיו זהים בכל הדפים המוצגים.

4.1.3 כרטיס מתאם למסך צבע CGA (Color Graphics Adapter)

כרטיס מתאם למסך צבעוני (CGA Adapter) מכיל זיכרון RAM בגודל 16KByte לשמירת הנתונים שיש להציג על גבי המסך: נתוני גרפיקה וטקסט. מתאם המסך מאפשר הצגה של 256 תווים שונים בו זמנית באמצעות תרגום ערך התו לתבנית התו המוצגת על המסך. תבניות אלו נמצאות ברכיב ROM שגודלו 8KByte. תרגום ערך התו לתבנית נעשה על ידי מחולל התווים.

4.1.3.1 הצגת תווים במתאם מסך CGA

איך מוצגים הנתונים במסך?

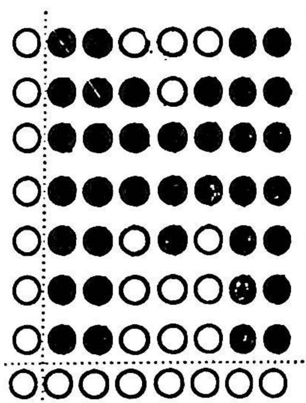
רזולוציה של מסך צבע CGA הינה 640 נקודות על ציר אופקי ו-200 נקודות על ציר אנכי. מכאן, שכאשר מציג המסך 80 תווים בשורה, ניתן להקצות 8 נקודות עבור רוחב התבנית של כל תו (640 נקודות מחולק ב-80 תווים). עבור אורך תבנית של כל תו ניתן להקצות 8 נקודות (200 נקודות מחולק ב-25 שורות). מכאן, שעבור הצגה של תווים מקצה מתאם המסך לכל תו תבנית (מטריצה) של 8x8 נקודות, אשר באמצעותה בונים את צורת התו. כאשר התבנית נשלחת למסך ניווכח שתמונת התו המוצג מורכבת מנקודות.

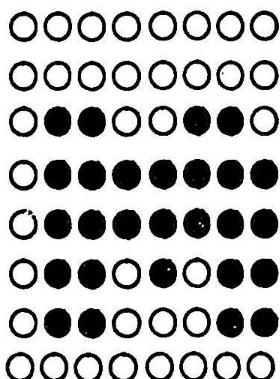
על מנת ליצור רווח בין השורות השונות ובין התווים בשורה, כוללת תבנית התו שורת נקודות אחת ריקה, אשר "נשתלת" בין השורות המוצגות. תווים אשר נאלצים לחרוג מתחת לקו התחתון של האותיות הגדולות באנגלית משתמשים בשורה תחתונה זו, כמו p למשל. לכן, נראה לעתים שתי אותיות הנמצאות זו מעל זו כשהן מחוברות ללא הפרדה של שורת רווח. תבנית התו כוללת גם עמודה אחת עבור רווח בין התווים המוצגים בשורה. כאן אפשר להעיר ששורת רווח זו מנוצלת גם עבור מספר אותיות בעברית אשר חורגות מתחום האות הסטנדרטי.

כתוצאה משמירה של שורת הרווח, גודל תבנית התו המשמשת להצגה הוא 7x7 נקודות ולא 8x8 נקודות. תבנית התו 7x7 במתאם מסך CGA מציגה את התווים באיכות הנמוכה ביותר מבין מתאמי המסך השונים, שבה ניתן לראות במסך את הנקודות המרכיבות את התו. תבנית בגודל 7x7 אינה מאפשרת להבליט את

הגופן ומקשה על ההפרדה בין התווים השונים כדי לתת צורה ייחודית לתו. היא גורמת לעייפות יתר של העיניים בעבודה ממושכת מול מסך CGA.

כאשר מציגים נתוני תווים (Text) מקציב מתאם המסך שני בתים בזיכרון RAM של המתאם עבור כל תו. בית ראשון מכיל את ערך התו שיש להציג (לפי טבלת ASCII) ובית שני מכיל את התכונות השונות (Attributes) שייתנו לתו כאשר יוצג על גבי המסך. תכונות אלו ניתנות עבור כל תו בנפרד והן מאפשרות על כן להציג כל התו בצבע רצוי, חוזק הארה (Intensity) והבהוב (Blink).





הצגת תווים במסך CGA

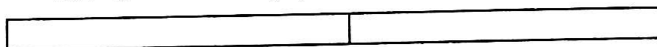
בהצגת טקסט קיימת אפשרות להציג 80 תווים בשורה ב-25 שורות (25x80) או 40 תווים בשורה ב-25 שורות (25x40). הפינה השמאלית העליונה בשני המקרים מסומנת (0,0). בהצגה 25x80 הפינה הימנית התחתונה תהיה (79,24) ובהצגה 25x40 פינה זו תהיה (39,24). שים לב שכמה משפות התכנות סימון הקואורדינטות הוא מ-1,1 עד 80,25.

במסך אפשר להציג 2000 תווי טקסט במערך של 25x80 ולכן צריך עבורם 4KByte זיכרון RAM. בהצגה של 25x40 שורות, אפשר להציג על המסך 1000 תווים וצריך זיכרון RAM בגודל 2KByte על מנת לשמור את המידע המוצג.

הכתיבה למתאם המסך

כאשר רוצים לכתוב את התווים ישירות ל-RAM, צריך לכתוב למתאם המסך שני בתים עבור כל תו שיוצג. שני בתים אלה מכילים את ערך התו שיש להציג ואת התכונות שלו. תהליך זה שקוף למשתמש, מכיון שהוא מקיש תו על גבי לוח המקשים והתו מופיע על גבי המסך.

בעת הצגת התו על גבי המסך, ערך התו מתורגם על ידי מחולל התווים לתבנית של נקודות המייצגת את תמונת התו. התכונות השונות שיקבלו הנקודות המרכיבות את התו (צבע, חוזק הארה, הבהוב) מתורגמות ומבוקרות על ידי הבית המכיל את תכונות התו. השימוש בשני בתים עבור כל תו מקנה יתרון מהירות למתאמי מסך הפועלים בממשק ברוחב 16 סיביות.



הקצאת מקום לכל תו במתאם

בית התוכן של התו קובע מה יוצג. יש 256 צורות שונות אפשריות לתו, אשר מוגדרות על ידי טבלת תווים בינלאומית הנקראת קוד ASCII (ראה נספח). טבלת ASCII מתאימה לכל ערך מבין 256 הערכים השונים של בית התוכן. התמונה מורכבת מתבנית (מטריצה) של נקודות הנקראת **גופן** (Font).

ערכי הסיביות בבית תכונות

7	6	5	4	3	2	1	0
הבהוב	צבע הרקע			חוזק הארה	צבע התו		
B	R	G	B	I	R	G	B

תפקיד הסיביות השונות בבית התכונות

הבית המשמש לתיאור התכונות של התו שיש להציג מכיל 8 סיביות שתפקידן:

* **צבע התו** – סיביות 0-2 מגדירות את צבע התו (Foreground color). צבע זה ניתן לבחירה מתוך 8 צבעי הצגה שונים (3 סיביות מאפשרות 8 צירופים שונים).
ראה להלן את הטבלה לבחירת הצבעים.

* **עוצמת הארה** – סיבית 3 מגדירה את עוצמת ההארה של התו שיוצג. יחד עם 3 סיביות (0-2) קודמות אפשר לבחור צבע תו מתוך 16 צבעים שונים. עוצמת ההארה נותנת גוון שונה לצבע ועל ידי כך מתוספים 8 צבעים.

* **צבע רקע** - סיביות 4-6 מגדירות את צבע הרקע (Background color) של התו המוצג. את צבע הרקע ניתן לבחור מתוך 8 צבעים שונים, מבין 8 הצבעים הראשונים בטבלת בחירת הצבעים. עבור כל תבנית תו ניתן לבחור גם את צבע הרקע של תבנית התו וכך ניתן להגדיל את מספר הצבעים המוצגים על גבי המסך. הצבעים הנוספים נוצרים בשל שילוב הצבעים בין צבע תבנית התו לבין צבע הרקע.

* **הבהוב** - סיבית 7 משמשת לבקרת הבהוב (Blink) של התו על גבי המסך. ערך של 1 בסיבית זו יגרום לתו להבהב בקצב קבוע על גבי המסך וערך 0 יגרום לתו להופיע באופן קבוע על גבי המסך. תכונה זו קשורה אך ורק לתו מסוים ולא לכל התווים במסך.

הצבע שיוצג		עוצמת הארה	צבע התו שיוצג R	G	B
Black	שחור	0	0	0	0
Blue	כחול	0	0	0	1
Green	ירוק	0	0	1	0
Cyan	כחול + ירוק	0	0	1	1
Red	אדום	0	1	0	0
Magenta	אדום + כחול	0	1	0	1
Brown	חום	0	1	1	0
White	לבן	0	1	1	1
Gray	אפור	1	0	0	0
Light Blue	כחול בהיר	1	0	0	1
Light Green	ירוק בהיר	1	0	1	0
Light Cyan	כחול + ירוק בהיר	1	0	1	1
Light Red	אדום בהיר	1	1	0	0
Light Mgenta	אדום+כחול בהיר	1	1	0	1
Yellow	צהוב	1	1	1	0
Light White	לבן מואר	1	1	1	1

טבלה לבחירת צבעים

נבחן לדוגמה, את המקרה הבא:

נתון תוכן של 2 בתים ב-RAM המכילים את התו שיש להציג ואת תכונותיו:

7	6	5	4	3	2	1	0
1	0	0	1	0	1	0	0

בית תכונות (N+1)

7	6	5	4	3	2	1	0
0	1	0	0	0	0	0	1

תו להצגה (N)

כתוצאה יוצג התו A, מכיון שהערך 41Hex בטבלת ASCII מייצג את התו A. תבנית התו תופיע בצבע אדום (ערך 1), צבע הרקע יהיה כחול+ירוק (ערך 4) והתו יהבהב באופן קבוע על גבי המסך (סיבית שמאלית 1).

4.1.3.2 הצגה של מספר דפים — Pages

כאשר מתאם המסך מציג טכסט במערך של 80x25 תווים, נדרש זיכרון של 4KByte בלבד מתוך 16KByte זיכרון RAM שהנמצא על הכרטיס המתאם, או 2KByte עבור מערך של 40x25 תווים. ניתן לנצל את זיכרון RAM הפנוי בכרטיס המתאם לשמירת מידע נוסף והצגתו על פי דרישה. באופן זה מחלקים את הזיכרון ל-4 חלקים בהצגת 80x25 תווים, או 8 דפים כאשר מציגים 40x25 תווים. כל חלק כזה, הקרוי **דף** (Page), מכיל קיבולת נתונים של מסך שלם, אשר ניתן להצגה באופן בלתי תלוי, כי הכתיבה נעשית בנפרד בכל אחד מהדפים.

בכל נקודת זמן, מופיעים על גבי המסך נתונים של דף אחד בלבד מבין הדפים שנמצאים במתאם. ניתן לכתוב נתונים לדף אחד בלבד, גם אם הוא אינו מוצג באותו זמן על המסך. על מנת לגרום למתאם המסך להציג דף כלשהו מבין הדפים האפשריים, התכנית שולחת פקודה למתאם המסך להחליף את הדף המוצג. החלפת הדפים נעשית בחומרה במהירות רבה וללא שהיות ועל כן המשתמש אינו רואה כיצד תמונת הדף, כלומר הנתונים, משתנית!

הדף המוצג על גבי המסך הוא **דף נראה** (Visible Page) והדף שאליו כותבים הוא **הדף הפעיל** (Active Page). עבור כל דף שומרת תכנית BIOS מידע על המיקום (X,Y) של הסמן (Cursor). בעת החלפה של הדפים היא מקפיצה את הסמן למקום על גבי המסך שבו הוא היה בפעם האחרונה כשדף זה הוצג.

4.1.3.3 אופני פעולה שונים של מתאם מסך CGA

לסיכום, מתאם מסך CGA יכול לפעול במספר אופנים (Modes):

BW80 - הצגה של 80 תווים ב-25 שורות הצגה בשני צבעים בלבד (ש/ל).
BW40 - הצגה של 40 תווים ב-25 שורות הצגה בשני צבעים בלבד (ש/ל).
CO80 - הצגה של 80 תווים ב-25 שורות. הצגה ב-16 צבעים שונים.
CO40 - הצגה של 40 תווים ב-25 שורות. הצגה ב-16 צבעים שונים.

4.1.3.4 הצגה של גרפיקה

הצגה של תמונה גרפית שונה בתכלית מהצגה של תווים. בהצגה גרפית מנוצל כל זיכרון RAM להצגה של המידע הגרפי. המסך מחולק **ליחידות תמונה** (Pixels), או נקודות, אשר המשתמש יכול לציין כל אחת מהן ולקבוע את צבעה. היכולת לקבוע את ערכה ומיקומה של כל נקודה על פני המסך מאפשרת לצייר צורות וקווים.

איכות ההצגה של התמונה הגרפית תלויה במושג הנקרא **הפרדה**, או **רזולוציה** (Resolution). הרזולוציה מגדירה את צפיפות הנקודות על גבי המסך. במתאם מסך CGA הרזולוציה המירבית היא 640x200 כלומר, 640 נקודות על ציר X האופקי ו-200 נקודות על ציר Y האנכי. המסך בנוי באופן שהנקודה השמאלית העליונה היא הנקודה 0,0 והנקודה הימנית התחתונה היא 639,199. ערכה של הנקודה הימנית התחתונה יכול להיות גם 319,199 או 159,99 על פי הרזולוציה שנבחרה להצגה. באופן פעולה גרפי, מתאם המסך תומך ב-3 אופני הצגה ברזולוציה שונה:

* **640 x 200** - באופן פעולה של רזולוציה גבוהה (High Resolution). מוצג המידע בשני צבעים בלבד: שחור ולבן. רזולוציה זו היא הגבוהה ביותר האפשרית במתאם, אך איכות ההצגה היא על חשבון מספר הצבעים שבהם ניתן לצבוע כל נקודה. באופן זה, לכל נקודה במסך מקצים סיבית אחת מתוך זיכרון RAM. מכיון שכל סיבית יכולה לייצג אחד משני צירופים בלבד (0 או 1) הרי שניתן לייצג שני צבעים בלבד (0 עבור נקודה שחורה ו-1 עבור נקודה לבנה).

  ברזולוציה זו כל בית בזיכרון מייצג ערך של 8 נקודות (Pixels). לדוגמה, אם נתכנת את מתאם המסך לאופן פעולה גרפי ברזולוציה של

640x200, הרי שכתיבת הערך 55Hex (55H=01010101B מספרי הסיביות משמאל לימין) לכתובת זיכרון B800:0Hex, נקודה ראשונה על גבי המסך (צד שמאל למעלה) תהיה שחורה (ערך הסיבית 0) שנייה לבנה (ערך הסיבית 1), שלישית שחורה (ערך 0) וכו'.

* **320 x 200** - ברזולוציית ביניים (Medium Resolution) מוצג המידע ב-4 צבעים בלבד. ניתן לבחור צבע אחד מתוך אוסף של 16 צבעים עבור הרקע (Pallete) ואחד מתוך 4 צבעים (שחור, ירוק, אדום, חום או שחור, לבן,...) עבור צבע ההצגה של כל נקודה. ברזולוציה זו מקצים לכל נקודה על גבי המסך 2 סיביות בתוך זיכרון RAM. ומכיון שבעזרת 2 סיביות ניתן לייצג 4 צירופים שונים (00, 01, 10, 11), הרי שכל צמד סיביות מייצגות אחד מתוך 4 צבעים עבור כל נקודה במסך.

כל בית בזיכרון RAM מייצג ערך של 4 נקודות (Pixels) מהמסך. לדוגמה, אם נתכנת את מתאם המסך לרזולוציה של 320x200, הרי שכתיבת הערך 1BHex (00011011B מספרי סיביות משמאל לימין) לכתובת B800:0 במרחב הזיכרון ונבחר אוסף צבעים מתוך ירוק, אדום וחום על רקע שחור, נקבל את התצוגה הבאה: נקודה ראשונה על גבי המסך בצבע שחור (ערך 00 מייצג צבע שחור), נקודה שנייה ירוקה (ערך 01 מייצג צבע ירוק), נקודה שלישית אדומה (ערך 10 מייצג צבע אדום) ונקודה רביעית חומה (ערך 11 מייצג צבע חום).

* **160 x 100** - באופן פעולה ברזולוציה נמוכה (Low Resolution) מוצג המידע הגרפי באחד מתוך 16 צבעים (לפי טבלת צבעים, כולל שחור ולבן) ברזולוציה הנמוכה ביותר. ברזולוציה זו מקצים לכל נקודה 4 סיביות בזיכרון RAM. מכיון שבעזרת 4 סיביות ניתן לייצג 16 צירופים שונים הרי שניתן להציג בכל נקודה אחד מתוך 16 צבעים שונים.

כל בית בזיכרון RAM מכיל מידע של 2 נקודות מהמסך (מספרי נקודות משמאל לימין). לדוגמה, אם נכתוב את הערך 1BHex (00011011B מספרי סיביות משמאל לימין) לכתובת זיכרון B800:0, הרי שנקודה ראשונה על גבי המסך תהיה כחולה והשנייה - Cyan בהיר.

בכל אופני הפעולה הגרפית, הזיכרון בכרטיס המתאם (B800:0) מחולק לשני חלקים. חלק אחד אחראי על השורות הזוגיות והחלק השני על השורות האי-זוגיות. כך שאם נרצה להאיר את חצי המסך התחתון בנקודות בצבע כלשהו, משמעות הדבר היא שנמלא את החצי התחתון בחלק ראשון של הזיכרון במתאם המסך ונמלא את החצי התחתון בחלק שני של זיכרון מתאם המסך. אם נמלא רק חלק תחתון של מתאם מסך נקבל נקודות מוארות אך ורק בשורות הזוגיות במסך והשורות האי-זוגיות תהיינה ריקות. כלומר, המסך יואר "שורה כן שורה לא". להלן תמונת הזיכרון של מתאם המסך עבור מקרה זה.

כתובת	
B800:0	שטח זיכרון המאחסן את כל השורות הזוגיות במסך בציר האנכי 0, 2, 4, 6, 8 ... 198
B800:1F3F	שטח זיכרון לא מנוצל
B800:2000	שטח זיכרון המאחסן את כל השורות האי זוגיות במסך בציר האנכי 1, 3, 5, 7, 9 ... 199
B800:3F3F	שטח לא מנוצל
B800:1FFF	

4.1.4 מתאם מסך מונוכרום MDA ומתאם הרקולס HGC

מתאם מסך MDA קיים רק במערכות מחשבים ישנות. מתאם זה זהה למתאם מסך HGC, אך אין לו יכולת הצגה גרפית. מכאן, שכל הנאמר בנושא גרפיקה מתאים אך רק למתאם מסך HGC, או מתאם מסך תואם HGC. כל ההסברים לגבי הצגת טקסט מתאימים לכל סוגי מתאמי מסך מונוכרום.

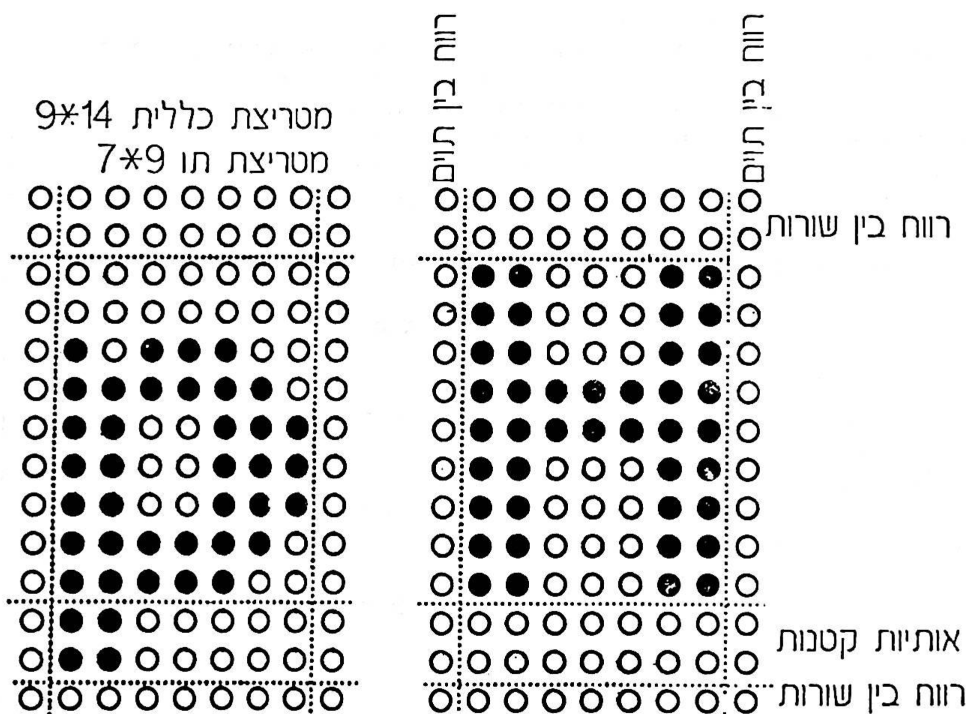
מתאם מסך מונוכרום יכול להציג אך ורק בשני צבעים. כתובת הזיכרון של כרטיס המתאם שונה מכתובת כרטיס מתאם מסך CGA, וכך גם כתובות מרחב הקלט/פלט של מתאם המסך שבתוך הכרטיס. כרטיס מתאם מסך MDA מכיל 4KByte של RAM בלבד. מתאם מסך HGC או תואם מכיל זיכרון RAM בגודל 32KByte ויכול להציג גם מידע גרפי (כדוגמת כרטיסי הרקולס, גנואה וכו').

4.1.4.1 הצגת תווים במתאם מונוכרום

הרזולוציה של מסך המונוכרום הינה 720 נקודות על הציר האופקי ו-350 נקודות על הציר האנכי. מכאן נובע שבצורת הצגה של 80 תווים בשורה ב-25 שורות, רוחב התבנית של כל תו הינה 9 נקודות (720 נקודות בשורה אופקית חלקי 80 תווים בשורה) וגובה התבנית הוא 14 נקודות (350 נקודות חלקי 25 שורות). כלומר, להצגה של תווים מקציב מתאם מסך המונוכרום תבנית (מטריצה) של 9x14 נקודות.

בעזרת תבנית נקודות זו בונים את צורת התו אשר נשלח למסך. בחלק עליון בתבנית מקצים שתי שורות כדי ליצור רווח בין השורות. שורה נוספת בתחתית התבנית מוקצית לרווח בין השורות. בנוסף, מקצים שתי שורות נוספות בתחתית התבנית עבור אותיות קטנות חורגות באנגלית כמו y או q למשל. עמודה אחת מכל צד בתבנית מוקצבת עבור רווח בין התווים המוצגים בשורה. שורות רווח והדגשה משאירות תבנית של 7x9 נקודות לכל תו ולא 9x14 נקודות שמהוות את התבנית הבסיסית. התבנית המוקצית לתמונת תו היא 7x11, אך עבור כל תמונת תו, התבנית שבשימוש היא 7x9.

תבנית התו במסך מונוכרום צפופה יותר מתבנית התו במסך CGA ולכן איכות ההצגה במסך מונוכרום טובה יותר, והגופן המגדיר את צורת האותיות והתווים יפה ומלוטש יותר. הליטוש ואיכות האות נובעים מכך, שככל שתבנית האות צפופה יותר ניתן להדגיש יותר פרטים בתמונת התו וליצור הבדלים חדים וברורים בין האותיות השונות.



תבנית תו במסך מונוכרום

בדומה למתאם מסך צבע CGA, גם מתאם מסך מונוכרום יכול להציג 256 תווים שונים בו-זמנית. ברכיב ROM (בגודל 8KByte) במתאם המסך נמצאות תבניות המתארות את תמונות של כל התווים. מתאם מסך מונוכרום אינו תומך בטכניקה של מיפוי דפים, למרות שבמתאם מסך מונוכרום בעל יכולת גרפית (HGC ותואמיו) קיים זיכרון ROM בגודל 32KByte, אשר יכול להתאים למטרה זו. שמירת נתוני התווים בזיכרון RAM שעל גבי הכרטיס זהה לאופן שבו נשמרים הנתונים בכרטיס CGA.

במתאם מסך מונוכרום מוקצים עבור כל תו שני בתים, המכילים את ערכו של התו ואת התכונה המתארת את האופן שבו הוא יוצג על המסך. מספר התכונות האפשריות עבור כל תו במתאם מסך מונוכרום מוגבל, בשל העדר יכולת הצגה בצבע. לכן, רק לחלק מהערכים שניתנו עבור תכונות תו במסך CGA יש משמעות במתאם מסך מונוכרום. הערכים האפשריים עבור תכונת התווים מובאים בטבלה הבאה:

הבהוב B	תכונת הרקע	חוזק הארה I	תכונת התו	
7	6 5 4	3	2 1 0	סיביות בבית התכונות

התכונה המייחדת את התו	I 2 1 0	B 6 5 4	ערך הקסה
תו שחור על רקע שחור-לא מוצג	0 0 0 0	0 0 0 0	0
תו לבן על רקע שחור	0 1 1 1	0 0 0 0	7 Hex
תו לבן על רקע שחור מואר	1 1 1 1	0 0 0 0	0F Hex
תו שחור על רקע לבן (Reverse)	0 0 0 0	0 1 1 1	70 Hex
קו מתחת תו לבן רקע שחור	0 0 0 1	0 0 0 0	1 Hex
קו מתחת תו לבן רקע שחור מואר	1 0 0 1	0 0 0 0	9 Hex
תווים מהבהבים על גבי המסך			
תו לבן על רקע שחור	0 1 1 1	1 0 0 0	87 Hex
תו לבן על רקע שחור מואר	1 1 1 1	1 0 0 0	8F Hex
תו שחור על רקע לבן (Reverse)	0 0 0 0	1 1 1 1	F0 Hex
קו מתחת תו לבן רקע שחור	0 0 0 1	1 0 0 0	81 Hex
קו מתחת תו לבן רקע שחור מואר	1 0 0 1	1 0 0 0	89 Hex

טבלה לבחירת תכונות תווים

4.1.4.2 הצגה גרפית באמצעות מתאם מסך מונוכרום HGC

לא קיימת אפשרות להציג מידע גרפי באמצעות מתאם מסך מונוכרום MDA המכיל זיכרון RAM בגודל 4KByte. הצגה של מידע גרפי אפשרית אך ורק במתאם מונוכרום עם יכולת גרפית המכיל זיכרון RAM בגודל 32KByte, כדוגמת כרטיס הרקולס - HGC (Hercules Graphics Card) וכרטיסים מתאמים תואמי הרקולס. הצגת של מידע גרפי שונה מהצגת מידע תווים בכך שמשתמשים בכל זיכרון RAM לשמירת הנתונים הגרפיים והצגתם על גבי המסך.

בהצגה גרפית מחולק המסך לנקודות בהתאם לרזולוציה. שליטה על הארה בכל נקודה מאפשרת לצייר צורות גרפיות שונות וליצור תמונה גרפית באיכות סבירה. איכות הצגת המידע הגרפי תלויה ברזולוציה של המסך.

הרזולוציה המירבית במתאם מונוכרום היא 720x348. ניתן אומנם להציג על המסך 350 שורות של נקודות, אך מכיון שנוח יותר לחלק את המסך ל-4 חלקים שווים, מנצלים אך ורק 348 שורות (348 מתחלק בצורה שלמה ב-4). מכאן, שבמסך ישנן 720 נקודות על ציר X האופקי ו-348 נקודות על ציר Y האנכי.

ערכה של כל נקודה גרפית על המסך יכול להיות שחור או לבן בלבד, ולכן כל בית של זיכרון RAM מכיל מידע עבור 8 נקודות שונות. לדוגמה, אם

הבית הראשון בזיכרון מכיל את הערך הבינארי **01010101**, הרי הנקודה הראשונה על המסך תהיה שחורה, הנקודה השנייה תהיה לבנה, השלישית שחורה וכו'. ציון הקואורדינטות במסך הגרפי: הנקודה השמאלית העליונה היא נקודה **0,0** והנקודה הימנית התחתונה היא **719,347**.

ההבדל היחיד לגבי צורת ההצגה הגרפית בין מתאם מסך צבע לבין מתאם מסך מונוכרום גרפי הוא החלוקה של הזיכרון לשורות הצגה. בעוד שזיכרון מתאם הצבע חולק לשני חלקים (שורות זוגיות ואי זוגיות), מחולק זיכרון מתאם המונוכרום ל-4 חלקים שווים על פי הפירוט הבא:

B000:0	שטח זיכרון המאחסן את כל השורות שמספרן אשר מחולק ב-4 נותן שארית 0 0, 4, 344 ....8 (88 שורות הצגה)
B000:1E96	מרחב זיכרון לא מנוצל
B000:2000	שטח זיכרון המאחסן את כל השורות שמספרן אשר מחולק ב-4 נותן שארית 1 1, 5, 345 ....9 (88 שורות הצגה)
B000:3E96	מרחב זיכרון לא מנוצל
B000:4000	שטח זיכרון המאחסן את כל השורות שמספרן אשר מחולק ב-4 נותן שארית 2 2, 6, 346 ....10 (88 שורות הצגה)
B000:5E96	מרחב זיכרון לא מנוצל
B000:6000	שטח זיכרון המאחסן את כל השורות שמספרן אשר מחולק ב-4 נותן שארית 3 3, 7, 347 ....11 (88 שורות הצגה)
B000:7E96	מרחב זיכרון לא מנוצל
B000:7FFF	

4.1.5 מתאם למסכי EGA ומעלה (VGA וכו')

מתאם מסך EGA (Enhanced Graphics Adapter) יכול להציג נתונים ב-16 צבעים שונים בדומה למתאם מסך CGA. הרזולוציה המקובלת של מסך EGA היא 640x350, שהינה טובה בהרבה מזו של מסך CGA. על פי תקן יבמ קיימים מסכי Extended EGA, או Super EGA ברזולוציה גבוהה יותר.

זיכרון RAM על גבי כרטיס מתאם EGA או VGA, גדול יותר מהזיכרון במתאם מסך CGA. כרטיס זה יכול לפעול באופן מושלם גם ככרטיס מתאם מונוכרום וגם ככרטיס מתאם ל-CGA.

מסכי EGA ומסכי VGA המשופרים יותר, הם המסכים המקובלים הטובים ביותר, בכך שהם משלבים רזולוציה גבוהה ואפשרות נרחבת להצגת צבעים. כיום נפוצים גם מסכים מסוג Super VGA, בעלי רזולוציה גבוהה יותר ויכולת הצגה של צבעים רבים יותר.

4.1.5.1 הצגת התווים

הרזולוציה במסך EGA היא 640 נקודות על ציר אופקי ו-350 נקודות על הציר האנכי של המסך. מכאן נובע, שהצגה של 80 תווים בשורה ב-25 שורות מאפשרת להקצות 8 נקודות (640 נקודות חלקי 80 תווים בשורה) עבור רוחב תבנית התו ו-14 נקודות (350 נקודות חלקי 25 שורות) עבור גובה תבנית התו. בהצגת תווים, התבנית המוקצית עבור כל תו היא, אם כן, 8x14 נקודות. תבנית בגודל זה מאפשרת להציג תווים באיכות הקרובה לאיכות ההצגה המושגת במתאם מסך מונוכרום, אך כאן נוסף גם הצבע.

מתאם מסך EGA תומך בטכניקה של מיפוי דפים ובאפשרותו להציג יותר מ-4 דפים האפשריים במתאם CGA. שמירת הנתונים בזיכרון RAM שעל הכרטיס זהה לאופן שמירתם בכרטיס CGA. גם כאן מוקצים עבור כל תו שני בתים המכילים את ערכו של התו ותכונותיו, אשר זהות לאלו של מתאם מסך CGA.

מתאם מסך EGA מאפשר לשנות את צורת התווים, כלומר, לשנות את הגופן שנקבע בטבלת ASCII. מחולל התווים במתאם מסך CGA, או במתאם מסך מונוכרום, יצייר על המסך את הגופנים (Fonts) על פי תוכנו של רכיב ROM שעל הכרטיס המתאם, אשר לא ניתן לשינוי תוך כדי פעולת המחשב. כדי להחליף גופנים צריך להחליף ROM, וזה מה שעושה היבואן של כרטיסי מסך CGA ו-MGA.

בזמן הדלקת המחשב שבו מותקן כרטיס מתאם EGA, נטענת משפחת הגופנים הראשונית מזיכרון ROM שעל הכרטיס המתאם אל זיכרון RAM של המתאם. מחולל התווים מקבל את תבניות הגופנים מזיכרון RAM ושולח אותן למסך. בזיכרון RAM של מתאם מסך EGA יש מקום ל-2 משפחות של גופנים שאפשר לטעון אותן לזיכרון על פי בחירת המשתמש. ניתן לעצב אותיות ותווים אחרים שאינם קיימים בטבלת ASCII ולהציג אותם באמצעות מתאם המסך. יצרני הכרטיס המתאם מספקים בדרך כלל תכנית המאפשרת לערוך ולטעון גופנים שונים על פי בחירתנו.

מערכת ההפעלה DOS כוללת משפחות גופנים סטנדרטיים, או Code Pages, אשר נקראות על פי ארצות שונות (ארה"ב, צרפת, אנגליה, ספרד, ישראל וכו'). מספר פקודות של מערכת ההפעלה נועדו לטפל בטעינת משפחת הגופנים למתאם

מסך EGA. אפשרות הטעינה של הגופנים ל-RAM פוטרת מהצורך להחליף את ה-ROM בכרטיס המתאם המתחבר למחולל התווים על מנת להחליף שפה בארצות שונות. בדרך זו נפתרה גם הוספת העברית. כזכור, הדבר אינו אפשרי בכרטיסי CGA ומונוכרום.

הכנסת עברית למתאם מסך EGA נעשית באמצעות טעינת תכנית בלבד! ניתן לכלול בקובץ AUTOEXEC.BAT פקודה כדוגמת תכנית EGAHE.COM אשר תטען בתהליך BOOT את הגופן המכיל עברית אל מתאם המסך. קיימים מתאמי מסך EGA שהיבואן החליף בהם את זיכרון ROM והכניס בו מראש את הגופן העברי. במתאמי מסך אלה אין צורך לטעון עברית בתוכנה.

עבור הצגת תווים, כרטיס מתאם מסך EGA ממופה כברירת מחדל באותן כתובות זיכרון כמו מתאם מסך CGA. כלומר, כל ההסברים שניתנו על הצגת תווים ואחסון תווים בזיכרון עבור מתאם מסך CGA, תקפים גם עבור מתאם מסך EGA. ההבדל הגדול הינו באיכות ההצגה המשופרת של התווים הנובעת מתבנית נקודות גדולה יותר להצגת הגופנים.

מן הראוי לציין כאן שהצגת תווים נעשית באופן דומה במסכים מסוג EGA ובמסכים מסוג VGA. ההבדל בין שני סוגי מסכים אלה קיים רק בגרפיקה.

4.1.5.2 גרפיקה

הגרפיקה במתאם מסך EGA משופרת יותר מאשר במתאמי מסך CGA ו-HGC, אך גם כאן נמצא מתאמי מסך בעלי תכונות שונות. מתאמי המסך הראשונים נבנו עם זיכרון בגודל 64KByte להצגת גרפיקה. זיכרון בגודל זה מאפשר תמיכה בשני אופני פעולה:

- רזולוציה 640x350, 2 סיביות לנקודה, 4 צבעים אפשריים.
- רזולוציה 640x200, 4 סיביות לנקודה, 16 צבעים אפשריים.

כרטיסי תיאום למסכי EGA מכילים כיום בדרך כלל 256KByte זיכרון RAM ואפשרות לפעול כמתאם מסך Super EGA. זיכרון בגודל זה מאפשר פעולה עם דפים גרפיים בדומה לפעולה עם דפים בהצגת טקסט. כאשר מתכנתים את מתאם מסך EGA לאופן פעולה גרפי, יש **דף פעיל** (Active), שאליו נשלח המידע מהמעבד, **ודף נראה** (Visible), שאת תוכנו מציג המתאם על גבי המסך.

כדי ליצור אפקט של גרפיקה מהירה, ניתן לכתוב מידע גרפי לתוך דף פעיל מספר 2, בזמן שדף 1 הוא הדף המוצג על המסך. עם סיום כתיבת המידע הגרפי ניתן להפוך את דף מספר 2 לדף נראה, אשר תוכנו יוצג על המסך ולכתוב מידע בדף מספר 1. פעולת המיתוג בין הדפים נמשכת מחזורי מכונה בודדים ועל כן החלפת הציור על המסך מהירה מאוד. ניתן לראות כי קיימים מספר אופני פעולה במתאם מסך EGA המכיל זיכרון RAM בגודל 256KByte:

- רזולוציה 640x200, 4 סיביות לנקודה, 16 צבעים, 4 דפים גרפיים.
- רזולוציה 640x350, 4 סיביות לנקודה, 16 צבעים, 2 דפים גרפיים.
- רזולוציה 800x400, 4 סיביות לנקודה, 16 צבעים, דף גרפי יחיד (אופן פעולה כמתאם מסך Super EGA).

אפשרות עבודה ברזולוציה 800x400 של Super EGA אינה נתמכת על ידי רוב התוכנות. על כן, הכרטיס אשר מאפשר אופן פעולה זה משווק עם ממשק

למרבית התכניות הגרפיות הנפוצות. כדי להפעיל את הכרטיס באופן פעולה Super EGA דרוש מסך MultiSync.

4.1.5.3 בחירת צבעים לתצוגה

ניתן להציג על מסך EGA ‏16 צבעים באופן פעולה גרפי, אבל באילו 16 צבעים נבחר לתצוגה? במתאם מסך EGA קיימת טבלת גוונים, אשר ממנה נלקח הצבע שאותו בוחר המשתמש. כשנבקש לצבוע נקודה כלשהי בצבע מספר 5, המתאם לוקח את הצבע מכניסה מספר 5 בטבלת הגוונים. בתהליך BOOT טבלת הגוונים מכילה את הגוונים שהוכנסו כברירת מחדל. ערכים אלה מתאימים לטבלת הגוונים הקיימת במתאם מסך CGA. ניתן לרשום בטבלה ערכים אחרים באמצעות פקודות מתאימות.

כתיבת ערכים חדשים לטבלה מאפשרת לשנות צבע מסוים לצבע אחר כלשהו, אך שינוי מסוג זה אינו מקובל (אפשר לעשות זאת גם במתאם מסך CGA). השימוש העיקרי נובע מכך שאפשר להכניס לטבלה ערכים רצויים כלשהם כדי ל"שחק" עם גווני צבע. לדוגמה, אפשר לכלול בטבלה 4 גוונים שונים של צבע צהוב ב-4 כניסות שונות (על חשבון 4 צבעים אחרים) ובעזרת גווני הצהוב לצייר על המסך דמות מוזהבת. את הערכים השונים שניתן לרשום בטבלה אפשר לבחור מתוך 64 גוונים שונים. במתאם מסך VGA ניתן לבחור 256 גוונים שונים, וזהו **ההבדל המהותי ביותר** בין EGA ל-VGA, מבחינת התוכנה.

מתאם מסך EGA ומעלה מחזיק 4 בנקים של זיכרון:
- בנק 1 לצבע ירוק,
- בנק 2 לצבע אדום,
- בנק 3 לצבע כחול,
- בנק 4 לעוצמת הארה.

במסך מוצגים ארבעת הבנקים בו זמנית. על כן, אם נדליק סיבית (נהפוך אותה מ-0 ל-1) בבנק 1 ונכבה אותה בשאר הבנקים, תתקבל על המסך נקודה ירוקה בפינה השמאלית העליונה של המסך. אם נדליק את הסיבית העליונה בכל הבנקים, נקבל נקודה לבנה בפינה הימנית התחתונה של המסך.

כיצד כותבים לבנקים השונים? כל הבנקים ממופים החל מכתובת A000:0 ואוגר מיוחד (3C5Hex) במרחב הקלט/פלט מנהל את הכתיבה אליהם. ארבע הסיביות התחתונות באוגר שייכות בהתאמה לארבעת הבנקים. לכן, הדלקת הסיבית התחתונה באוגר ולאחר מכן כתיבה למרחב הזיכרון A000, תצייר בירוק. הדלקת כל ארבע הסיביות (FHex) ולאחר מכן כתיבה למרחב הזיכרון A000 תבצע כתיבה בו זמנית לכל ארבעת הבנקים, כלומר, תצייר בלבן על המסך.

אנו ממליצים לבצע גרפיקה באמצעות שפת תכנות עילית בלבד. בכל זאת נזכיר, שניתן לכתוב לזיכרון המופיע בכתובת A000:0000Hex. הנקודות במסך מיוצגות בזיכרון RAM ברצף: בית ראשון בזיכרון 8 נקודות ראשונות, בית שני בזיכרון 8 הנקודות הבאות וכו'. כלומר, כדי להדליק נקודה לבנה על המסך עליך להפוך את הסיבית המתאימה ל-1. לדוגמה, על מנת להדליק נקודה לבנה בקצה העליון השמאלי במסך יש לכתוב את הערך 80Hex לכתובת A000:0Hex. כדי להדליק נקודה לבנה בקצה התחתון השמאלי של המסך יש לרשום את הערך 1 לכתובת A000:65DFHex (ראה תכנית דוגמה SCRGDEMO בנספח תכניות הדוגמה).

4.1.6 סיכום תכונות של מתאמי מסך ומסכים

מסך מונוכרום עם מתאם מסך HGC הוא טוב לפעולה, נוח לקרוא בו ואיכות ההצגה שלו סבירה. החיסרון הגדול של מסך מונוכרום הוא העדר יכולת הצגה בצבעים ולכן, מספר משחקים לא יפעלו עם מתאם מסך כזה. הפתרון לחוסר התאימות למשחקים הוא הוספת כרטיס מתאם CGA. ניתן להשיג כרטיס מתאם שמאפשר לפעול הן באופן HGC והן באופן CGA. תכניות המחייבות צבע תוכלנה לפעול על מסך מונוכרום, גם אם התוצאות לא תהיינה בדיוק מה שאנו רוצים: נקבל תחום של גווני הצבע האחד של המסך ולא צבעים שונים.

מסך צבע עם מתאם מסך CGA טוב למשחקים. אבל חלק מהמשחקים החדשים היום דורשים מסך EGA כמינימום. מסך CGA יקר יותר ממסך מונוכרום. יתרונו הגדול של מסך CGA הוא הצבע, ואין חולק על כך שהצבע יפה ומרשים יותר מהצגה במסך חד-צבע. חסרונו הגדול של מסך CGA הוא הרזולוציה הנמוכה שלו. כאשר משתמשים בו לעיבוד תמלילים או תכנות, גורמת הרזולוציה הנמוכה לעצבנות ולעייפות יתר של העיניים. ככלל, מומלץ לרכוש מסך מונוכרום או מסך EGA ומעלה, ולא מסך CGA.

מסך EGA או משופר יותר עם כרטיס מתאים עדיפים על מסכי CGA ומונוכרום. מסך EGA ומעלה משלב את יתרונות שני המסכים הקודמים: רזולוציה גבוהה בתוספת הצגה בצבעים. חסרונו הגדול של מסך זה הוא המחיר, גם אם מחירם של המסך והמתאם הולכים ויורדים.

מסך יבמ/8514 ומסכי רזולוציה גבוהה אחרים הינם הטובים ביותר,אך רוב התכניות הנפוצות אינן תומכות באפשרויות המתוחכמות של מסכים אלה. המחיר שלהם גבוה בהרבה ממחיר מסכי EGA ו-VGA. מתי אם כן דרוש מסך כזה? כאשר מייעדים מחשב כתחנת עבודה גרפית (Graphics Work Station) ויודעים שהיישומים שיופעלו בתחנה תומכים במסך כזה.

מסכים מתקדמים אלה משמשים בעיקר לתכניות שרטוט מתקדמות, כמו רוב תכניות השרטוט ההנדסי (CAD), אשר מצדיקות את המחיר. יש לציין את חסרונו של מסך 8514 ביחס לשאר המסכים מסוג זה: הוא פועל בשיטה של שורות זוגיות ואי-זוגיות לכל תמונה (כמו בטלוויזיה), אשר יוצרת הבהוב מסוים על גבי המסך. מסך 8514 מכיל מעבד גרפי המופעל באוסף פקודות מיוחד משלו.

נתמך ע"י מתאם מסך מסוג	מספר דפים	מספר צבעים אפשריים בו-זמנית במסך אחד	רזולוציה של המסך	גודל התו בפיקסל	כתובת הסגמנט בזיכרון	מספר עמודות ושורות של תווים	סוג כתיבה למסך	קוד אופן הפעולה של תכנית BIOS
דואלי, CGA	8	16	320*200	8*8	B800	40*25	טקסט	0, 1
דואלי, CGA	8	16	320*350	8*14	B800	40*25	טקסט	0, 1
EGA+, דואלי, CGA	8	16	360*400	9*16	B800	40*25	טקסט	0, 1
דואלי, CGA	8	16	640*200	8*8	B800	80*25	טקסט	2, 3
דואלי, CGA	8	16	640*350	8*14	B800	80*25	טקסט	2, 3
EGA+	8	16	720*400	9*16	B800	80*25	טקסט	2, 3
דואלי, CGA	1	4	320*200	8*8	B800	40*25	גרפי	4, 5
דואלי, CGA	1	2	320*200	8*8	B800	80*25	גרפי	6
דואלי, HGA	8	4	720*350	9*14	B000	80*25	טקסט	7
VGA+	8	4	720*400	9*16	B000	80*25	טקסט	7
EGA+	8	16	320*200	8*8	A000	40*25	גרפי	D
EGA+	4	16	640*200	8*8	A000	80*25	גרפי	E
EGA+	2	4	640*350	8*14	A000	80*25	גרפי	F
EGA+	1	16	640*350	8*14	A000	80*25	גרפי	10
VGA+	1	2	640*480	8*16	A000	80*30	גרפי	11
VGA+	1	16	640*480	8*16	A000	80*30	גרפי	12
VGA+	1	256	320*200	8*8	A000	40*25	גרפי	13
Super VGA+	2	16	1056*352	8*8	B800	132*44	טקסט	22
Super VGA+	4	16	1056*350	8*14	B800	132*25	טקסט	23
Super EGA+	1	16	640*480	8*8	A000	80*60	גרפי	25,26
Super VGA+	4	4	1188*350	9*14	B000	132*25	טכסט	50
Super VGA+	2	4	1188*352	9*8	B000	132*44	טכסט	52
Super EGA+	1	256	640*400	8*16	A000	80*25	גרפי	60
Super EGA+	1	256	640*480	8*16	A000	80*30	גרפי	61
Super EGA+	1	256	800*600	8*8	A000	100*75	גרפי	62
Super VGA+	1	256	1024*768	8*16	A000	128*48	גרפי	63
Super EGA+	1	16	800*600	8*8	A000	100*75	גרפי	6A,71
Super VGA+	1	16	1024*768	8*16	A000	128*48	גרפי	74
Super VGA+	1	4	1024*768	8*16	A000	128*48	גרפי	75
Super VGA+	1	2	1024*768	8*16	A000	128*48	גרפי	76

הערות: אופני פעולה 0-13Hex הם אופני פעולה תקניים של יבמ.
אופני הפעולה המיוחדים של Super VGA ו-Super EGA אינם מקיימים בהכרח תאימות מלאה במסכים שונים.

השוואת תכונות מסכים

4.1.7 הוראות לאופני פעולה במסכים

4.1.7.1 פקודת DOS: MODE

פקודה זו מאפשרת מעבר בין אופני הפעולה השונים של מתאם המסך. מתאמי המסך השונים תומכים באופני פעולה שונים, על מנת לשמור על תאימות כלפי תכניות שונות. מתאם מסך EGA, יכול לפעול גם כמתאם מונוכרום או CGA.

למשל, הפקודה MODE BW80
תעביר מתאם המסך CGA, EGA, או VGA לפעולה במצב של מתאם מסך מונוכרום כתוצאה, המתאם יגיב לתכונות תווים Attributes כמתאם מסך מונוכרום.

למשל, הפקודה MODE CO80
תחזיר את המתאם להצגה בצבעים על פי תכונות של מתאם מסך צבע.

4.1.7.2 קובץ תכנית ANSI.SYS

קובץ זה הינו תוכנית הפעלה להתקן (Device Driver) המשמשת לתיאום בין תכנית המתבצעת על גבי המחשב האישי, לבין תכניות שפותחו על גבי מחשבים אחרים. במחשב האישי מתאם המסך הוא חלק בלתי נפרד מהמערכת, אך במחשבים בינוניים וגדולים (כדוגמת VAX של דיגיטל) אין הדבר כך. המחשב תומך במספר גדול של תכניות ומשתמשים וכל משתמש פועל מולו בעזרת מסוף המכיל מסך, מקלדת, כרטיס מתאם למסך וכרטיס תקשורת לקישור אל המחשב המרוחק.

תכנית הרוצה להציג נתוני תווים במסוף המחובר למחשב שולחת אליו את ערכי התווים. כדי לציין את תכונותיהם משתמשים בצירופים של **תווי בקרה** (Escape Sequence). התו הראשון הוא תו בקרה מיוחד (Escape) אשר מסמן למתאם המסך כי יש לפענח את צירוף התווים שאחריו כתווי הוראות ולבצע את הפעולה המתבקשת מהם (שינוי צבע, חוזק הארה וכו'). תווי הבקרה אינם מוצגים על המסך.

התקן הבינלאומי לצירופים השונים של תווי הבקרה הללו נקרא תקן ANSI, על שם ועדת התקינה בארה"ב. ליצרנים שונים קיימים צירופים ייחודיים, אך בדרך כלל כל מסוף תומך גם בתקן ANSI. אם יצרן פיתח תוכנה במחשב גדול ורוצה לפתח גירסה עבור המחשב אישי, עליו להתאים את פקודות הבקרה המכוונות למסך. הפקודות המיועדות להצגת נתונים במסוף לא תפעלנה בצורה נכונה על מסך המחשב האישי, מכיון שכל הצגת התווים נעשית דרך התכנית הכתובה ב-ROM BIOS. תכנית BIOS אינה מכירה את תקן ANSI ולא מתייחסת לתו הבקרה (Escape) כתו מיוחד, אלא מציגה אותו ואת כל הבאים אחריו על גבי המסך. לדוגמה, הרצף "Escape [2 J" יגרום למחיקת המסך הפועל על פי תקן ANSI, כמו שפקודת CLS פועלת תחת מערכת ההפעלה DOS.

מפתחי תוכנה שאינם רוצים לשנות את כל פקודות הבקרה למסך מתוך חיסכון בזמן הפיתוח, יכולים להגדיר בקובץ CONFIG.SYS פקודה לטעינת תכנית ANSI.SYS אשר מבקרת את פעולת המסך. היא מאפשרת למחשב האישי להכיר את צירופי הבקרה השונים (Escape Sequence), לבצע אותם ולהציג על המסך את הנתונים בדיוק באופן שהיו מוצגים, אילו היה זה מסוף התומך בתקן ANSI. אם מפעילים תכנית המחייבת שימוש ב-ANSI צריך להגדיר פקודה זו בקובץ CONFIG.SYS:

```
DEVICE=C:\DOS\ANSI.SYS
```

4.1.7.3 גופנים סטנדרטיים (Code Pages)

מערכת ההפעלה DOS תומכת בהחלפת גופנים תוך כדי פעולה שוטפת. תמיכה זו נועדה לאפשר למשתמשים לפעול עם **דפי קוד סטנדרטיים** (Code pages), או גופנים של ארצות שונות, אשר כוללים תווים שונים ומבנה שונה של המקלדת. מתאמי מסך EGA ו- VGA כוללים גופנים הצרובים ב-ROM בהתאם לארץ היעד של המוצר. מערכת ההפעלה מאפשרת להגדיר את הגופנים שיוצגו על גבי המסך בהתאם לצירוף הגופנים ופריסת המקלדת האפשריים לבחירה בכל ארץ. פקודות טעינת גופנים מחייבות שימוש במתאם מסך EGA, או VGA.

על מנת להכין את המחשב האישי לשינוי גופנים ולהגדרת המקלדת על המשתמש לבצע הכנות מסוימות כפי שמפורט להלן.

קובץ CONFIG.SYS

בקובץ זה יש להגדיר:

* קוד "ארץ", אשר זהה לקידומת הארץ במערכת הטלפון הבינלאומית. לארצות שונות צירופי פונטים מיוחדים לתצוגה ולמערך התווים במקלדת, הצגת תאריכים וזמן באופן מסוים, הצגת מספרים וכו'. בקובץ COUNTRY.SYS כלולים ערכים אלה עבור הארצות השונות.

* טעינת תכנית הפעלה להתקן (Device Driver), אשר מטפלת בשינוי גופנים ובהגדרת סוג המסך (כזכור, רק מתאמי מסך EGA, VGA ומעלה תומכים בשינוי גופנים). תכנית זו היא התכנית DISPLAY.SYS.

* הרצת התכנית אשר טוענת ל-RAM את כל המידע על הארץ המבוקשת לצורך פעולה עם שינוי דפים.

הגדרות בקובצי CONFIG.SYS ו-AUTOEXEC.BAT

לצורך הדוגמה ניקח את המקרה הבא: גופנים אפשריים עבור ארה"ב הם 437 ו-850, קוד הארץ הוא 001 וקוד המקלדת הוא US. בהנחה שמתאם המסך במערכת הינו EGA, נשתמש בפקודות הבאות כדי להכין את המערכת לפעולה עם גופנים, או דפי קוד סטנדרטיים (Code Pages).

מניחים שידוע שהגופן ב-ROM של המתאם הוא 437. לפיכך,

בקובץ CONFIG.SYS נכלול את הפקודות הבאות:

```
COUNTRY=001,,C:\DOS\COUNTRY.SYS                          (1
DEVICE=C:\DOS\DISPLAY.SYS CON=(EGA,437,1)                (2
INSTALL=C:\DOS\NLSFUNC.EXE                               (3
```

בקובץ AUTOEXEC.BAT נכלול את הפקודות הבאות:

```
KEYB US,, C:\DOS\KEYBOARD.SYS                            (4
MODE CON: CODEPAGE PREPARE=((850)C:\DOS\EGA.CPI)         (5
```

הסבר:

1) בחירת הארץ, המידע נמצא בקובץ COUNTRY.SYS.
2) הגדרת המסך EGA. המספר 437 מציין שהגופן הבסיסי ב-ROM הינו 437, ו-1 מציין שניתן לטעון גופן אחד נוסף לגופן הקיים ב-ROM של הכרטיס.
3) טעינת המידע הייחודי לארץ המוצא.
4) שינוי פריסת המקלדת לפי ארה"ב (ברירת מחדל - לא מחייב).
5) טעינת הגופן הנוסף 850 לזיכרון של המתאם.

בשלב זה ניתן לעבור מגופן אחד למשנהו. צורת המעבר בין הגופנים:

כדי להציג את הגופן הפעיל יש לרשום את הפקודה CHCP

כדי לעבור לגופן 850 יש לרשום את הפקודה CHCP 850

מניחים שלא ידוע מהו הגופן ב-ROM של המתאם. לפיכך,

בקובץ CONFIG.SYS נכלול את הפקודות הבאות:

```
COUNTRY=001,,C:\DOS\COUNTRY.SYS                (1
DEVICE=C:\DOS\DISPLAY.SYS CON=(EGA,,2)         (2
INSTALL=C:\DOS\NLSFUNC.EXE                     (3
```

בקובץ AUTOEXEC.BAT נכלול את הפקודות הבאות:

```
KEYB US,, C:\DOS\KEYBOARD.SYS                              (4
MODE CON: CODEPAGE PREPARE=((850,437)C:\DOS\EGA.CPI)       (5
```

הסבר:

1) בחירת הארץ. המידע נמצא בקובץ COUNTRY.SYS.
2) הגדרת המסך EGA. המספר 2 מציין שניתן לטעון שני גופנים לזיכרון המתאם ושלא נשתמש בגופן הקיים ב-ROM של הכרטיס. יש לזכור, כי בכל פעם שנבצע BOOT למחשב, המתאם מכניס כברירת מחדל את הגופן הקיים ב-ROM שלו, ורק הפקודות שנגדיר ישנו זאת.
3) טעינת המידע הייחודי לארץ המוצא.
4) שינוי פריסת המקלדת לפי ארה"ב (ברירת מחדל - לא מחייב).
5) טעינת הגופנים הנוספים 850 ו-437 לזיכרון של המתאם.

בשלב זה ניתן לעבור מגופן אחד למשנהו. צורת המעבר:

כדי להציג את הגופן הפעיל יש לרשום את הפקודה CHCP

כדי לעבור לגופן 850 יש לרשום את הפקודה CHCP 850

כדי לעבור לגופן 437 יש לרשום את הפקודה CHCP 437

במעבר בין גופנים נראה הבזק קצר של המסך. אותיות וסימנים שהוצגו על גבי המסך וייצוגם שונה בין הגופנים השונים ישתנו (במעבר מגופן 437 ל-850, אותיות העברית יתחלפו באותיות בעלות צורות חסרות משמעות).

במערכת ההפעלה DOS כלולים מראש דפי קוד (Code Pages) סטנדרטיים אלה:

437 - עבור ארה"ב.
850 - בינלאומי, מתאים לרוב הארצות דוברות אנגלית (אירופה, צפון-אמריקה, דרום אמריקה וכו')
860 - עבור פורטוגל.
863 - עבור צרפת / קנדה.
865 - עבור מדינות סקנדינביה (שבדיה, דנמרק, נורווגיה).

דפי קוד אחרים זמינים לפי ארץ היעד של התוכנה. כך למשל בישראל זמין דף קוד 972.

4.1.8 גישה לכרטיס המסך דרך תכנית ROM BIOS

אפשר לכתוב נתונים על המסך באופנים אחדים:

א. כתיבה ישירות לכרטיס מתאם המסך. כתיבה בדרך זו הינה המהירה ביותר האפשרית בשילוב עם כתיבה בשפת אסמבלי, אך מחייבת את המתכנת לשלוט בצורה מירבית בכרטיסי החומרה השונים.

ב. פסיקת תוכנה הקוראת לתכנית השירות הנמצאת ברכיב ROM BIOS ומאפשרת להציג נתונים על גבי המסך ולשלוט על פעולת מתאם המסך. פסיקת ROM BIOS מתרגמת את הפרמטרים השונים שנשלחו אליה לפקודות המכוונות לכרטיס מתאם המסך, ולכן זמן הביצוע ארוך יותר מכתיבה ישירה למתאם.

ג. פסיקת תוכנה הקוראת לתכנית שירות של מערכת ההפעלה. מערכת ההפעלה מתרגמת את הנתונים והפרמטרים השונים הנשלחים על ידי התכנית הקוראת לצורה שבה הם צריכים להישלח לפסיקת השירות של ROM BIOS, מבצעת פסיקת תוכנה וקוראת לתכנית ROM BIOS. משך זמן הביצוע של פסיקת השירות של מערכת ההפעלה ארוך יותר ממשך זמן הביצוע הישיר דרך תכניות ROM BIOS, משום שיש צורך בתרגום כפול של הנתונים שיש להציג על גבי המסך: תרגום ראשון בתכנית השירות של מערכת ההפעלה DOS, ותרגום שני על ידי תכנית ROM BIOS.

ד. באמצעות שפת התכנות שבה משתמשים. בשפת אסמבלר יש לבצע כתיבה באחת מהשיטות הראשונות, אך בשפות עיליות כמו פסקל, או C, ישנם כלים נוחים לכתיבה למסך. חלק מהמהדרים משתמש בתכניות שירות של מערכת ההפעלה על מנת להציג על גבי המסך וחלקם קורא ישירות לתכניות ROM BIOS. מבחינת המתכנת המתחיל, כתיבה בדרך זו היא הנוחה והבטוחה ביותר.

בהמשך נציג את השירותים העיקריים שמספקת תכנית השירות של ROM BIOS על מנת להתקשר למתאם המסך, להציג נתונים ולבקר את פעולתו. הגישה לתכנית השירות לטיפול במסך היא באמצעות פסיקת תוכנה 10Hex. את הנתונים, או הפרמטרים, לפסיקת השירות יש להעביר באמצעות האוגרים של המעבד.

4.1.8.1 פסיקה 10Hex של BIOS לגישה למסך

פסיקה 10Hex של BIOS מספקת למתכנת את השירותים שהוא זקוק להם לצורך גישה וניהול המסך. ממשק הפקודות מורה לו מה צריך להכניס לאוגרים השונים (AL, AH וכו') כדי לקבל שירות מבוקש. פירוט האוגרים שיש להעביר לפסיקת השירות נמצא תחת הכותרת "**בקריאה לפסיקה**". כאשר פסיקת השירות מסיימת את הפעולה היא מחזירה לתכנית הקוראת ערכים המתארים: קודי שגיאה, פרמטרים מבוקשים וכו'. הערכים המוחזרים מפורטים תחת הכותרת "**בחזרה מפסיקה**".

פונקציות כלליות של 10Hex

0 - <u>כיוון אופן הפעולה של מתאם מסך (Mode)</u>

בקריאה לפסיקה:

AH = 0 כיוון אופן פעולה של מתאם המסך.

AL = ערך אשר מציין את אופן הפעולה המבוקש בהתאם לטבלה שלהלן:

אופן פעולה Video Mode (*)	צבעים	גרפיקה/ טקסט	רזולוציית הצגה שורות x עמודות (Y x X)	CGA	MONO HGC	EGA	VGA
00	2	טקסט	40 x 25	√		√	√
01	16	טקסט	40 x 25	√		√	√
02	2	טקסט	80 x 25	√		√	√
03	16	טקסט	80 x 25	√		√	√
04	4	גרפיקה	320 x 200	√		√	√
05	2	גרפיקה	320 x 200	√		√	√
06	2	גרפיקה	640 x 200	√		√	√
07	2	טקסט	80 x 25		√	√	√
0D	16	גרפיקה	320 x 200			√	√
0E	16	גרפיקה	640 x 200			√	√
0F	2	גרפיקה	640 x 350			√	√
10	16	גרפיקה	640 x 350			√	√
11	2	גרפיקה	640 x 480				√
12	16	גרפיקה	640 x 480				√
13	256	גרפיקה	320 x 200				√

* ראה טבלת אופני פעולה והשוואת תכונות מסכים.

אופני פעולה שונים של מתאמי מסך

בחזרה מפסיקה:

לא מוחזר ערך כלשהו בגמר פעולת השינוי.

1 - כיוון הצורה והגודל של הסמן (Cursor)

- הסמן מוצג על ידי החומרה באופן פעולה Text Mode בלבד.
- הסמן מוצג תמיד כמרובע שניתן להגדיר את גובהו.

בקריאה לפסיקה:

AH	=	1 כיוון צורה וגודל של הסמן.
CL	=	מספר שורה ראשונה בתבנית התו שבה יתחיל הסמן.
CH	=	מספר השורה בתבנית התו שבה יסתיים הסמן.

רוחב הסמן קבוע ובאמצעות פקודה זו ניתן לשלוט על גובהו במסגרת תבנית התו. השורה המירבית האפשרית במטריצת הנקודות של התו שבה יסתיים הסמן תלויה בסוג מתאם המסך ונקבעת על פי הגודל המוקצה לתו בכל סוג מסך. לדוגמה: במתאם מסך מונוכרום, שורה אחרונה אפשרית היא 13. במתאם מסך CGA שורה אחרונה אפשרית היא 7. מספרי השורות מתחילים במספר 0.

כרטיס החומרה של המסך הוא האחראי על הבהוב הסמן ואין למשתמש/מתכנת שליטה על כך.

בחזרה מפסיקה:
לא מוחזר ערך כלשהו באוגרים בגמר השינוי של צורת הסמן.

2 - קביעת מקום הסמן על גבי המסך

בקריאה לפסיקה:

AH	=	2 קביעת מקום הסמן. לאחר פקודה זו הסמן יופיע במקום אשר נקבע עבורו בפקודה.
BH	=	מספר דף. פרמטר זה ישים רק במתאמי מסך המאפשרים פעולה על מספר דפים (פרט למתאם HGC כל מתאמי המסך מאפשרים פעולה על דפי TEXT). הערכים האפשריים עבור אוגר AL הם 0 עד 7, שהם מספר הדפים האפשרי. לכל דף קיים סמן נפרד שניתן לשלוט על מיקומו.
DH	=	השורה בה יופיע הסמן (ציר Y).
DL	=	העמודה בה יופיע הסמן (ציר X).

הפקודה תפעל אך ורק באופן פעולה TEXT של מתאם המסך. הציון של הפינה השמאלית העליונה של המסך הוא 0,0 והציון של הפינה הימנית התחתונה הינו על פי אופן הפעולה הנבחר: 39,24 באופן פעולה של 40x25, או 79,24 באופן פעולה 80x25).

בחזרה מפסיקה:
לא מוחזר ערך כלשהו באוגרים בסיום הפעולה.

3 - קריאה והחזרה של מקום וצורת הסמן

בקריאה לפסיקה:

AH = 3 החזרה של מקום וצורת הסמן כפי שהוא מופיע על גבי המסך.
BH = מספר הדף שממנו יש להחזיר את מקום הסמן. לכל דף קיים סמן שניתן לשלוט בו בנפרד.

בחזרה מפסיקה:

DH = מספר השורה בה נמצא הסמן.
DL = מספר העמודה בה נמצא הסמן.
CH = השורה שבה מתחיל הסמן (צורת הסמן - ראה פקודה 1).
CL = השורה שבה מסתיים הסמן (צורת הסמן - ראה פקודה 1).

4 - קריאה של מיקום עט האור

עט אור (Light Pen) הוא התקן חומרה שניתן לחבר למסך באמצעות התקן מיוחד לטיפול בעט והוא מאפשר לתת פקודות למחשב באמצעות הצמדה של העט למסך והארה. מכשיר זה אינו נפוץ במיוחד, אך תכנית ROM BIOS מספקת כלים לטפל בו.

בקריאה לפסיקה:

AH = 4 קרא מיקום עט אור והחזר באוגרים את מיקומו.

בחזרה מפסיקה:

AH = 0 עט אור אינו פעיל ואין מידע נוסף על מיקומו.
1 המתג בעט האור לחוץ או פעיל ושאר האוגרים מחזירים את מיקומו של העט במסך.
BX = העמודה במסך שעליה מאיר עט האור (נקודה מסוימת על פי החלוקה הגרפית של המסך).
CX = השורה במסך שעליה מאיר עט האור (נקודה מסוימת על פי החלוקה הגרפית של המסך).
DL = עמודת TEXT שאותה מסמן עט האור.
DH = שורת TEXT שאותה מסמן עט האור.

5 - בחירת דף אקטיבי להצגה על גבי המסך

AH = 5 בחירת דף TEXT מבין הדפים הקיימים כדי שיופיע על המסך. שאר הדפים נסתרים באותו זמן מעיני המשתמש, אך תוכנם אינו נמחק. הדבר מספק למתכנת אפשרות לרשום נתונים בכמה דפים ולהחליף תצוגה במהירות גבוהה ללא כתיבה למתאם.
AL = מספר הדף שיוצג על גבי המסך. על פי אופן הפעולה וסוג מתאם המסך.

בחזרה מפסיקה:

הדף שנבחר יופיע על גבי המסך. הסמן של דף זה יופיע במקום שבו היה לאחר הכתיבה האחרונה לדף זה.

6 - גלגול המסך המוצג כלפי מעלה

7 - גלגול המסך המוצג כלפי מטה

בקריאה לפסיקה:

AH = 6 גלגל (Scroll) את תוכן המסך כלפי מעלה.
AH = 7 גלגל (Scroll) את תוכן המסך כלפי מטה.
AL = 0 גלגל את כל המסך מעלה/מטה. (תוכן המסך ייעלם).
AL מספר המציין כמה שורות יש לגלגל כלפי מעלה/מטה.
CL = עמודה התחלתית שמאלית של משטח הגלגול.
CH = שורה התחלתית עליונה של משטח הגלגול.
DL = עמודה ימנית תחתונה של משטח הגלגול.
DH = שורה תחתונה של משטח הגלגול.
BH = מציין מה תהיה התכונה של תוכן החלון שיתרוקן/יגולגל.

הסבר:

ערכי האוגרים CX ו-DX משמשים להגדרת החלון שיש לגלגל כלפי מעלה/מטה. השורות שמתחת לחלון יעלו/ירדו כלפי מעלה/מטה ומתחת/מעל ייכנס תוכן ריק (ערך 20Hex). הצבע והתכונות של כל תו שייכנס נקבעים על פי ערכו של אוגר BH. פקודת הגלגול שימושית בעיקר כשרוצים ליצור חלונות ולאפשר גלגול אך ורק של החלון ולא של כל המסך.

0F - החזרת אופן פעולה של מתאם המסך

בקריאה לפסיקה:

AH = 0FHex החזר את אופן הפעולה הנוכחי שבו נמצא מתאם המסך.

בחזרה מפסיקה:

AL = אופן הפעולה של מתאם המסך (ראה את הפקודה 0 כדי לכוון את אופן הפעולה של מתאם המסך).
AH = מספר עמודות על גבי המסך.
BH = מספר הדף המוצג על גבי המסך.

פונקציות לטיפול בתווים של פסיקת 10Hex

הפקודות הבאות תפעלנה בצורה מלאה וללא תוספות רק באופן פעולה TEXT. אם המתאם נמצא באופן גרפי, פסיקות אלו תפעלנה בצורה חלקית. כדי לשפר מצב זה קיימת פסיקה 1FH וכדי לטפל בה היעזר בספר: DOS Technical Reference.

8 - החזרת ערך התו ותכונתו ממסך

בקריאה לפסיקה:

AH = 8 החזר את ערך התו מהמקום שבו מופיע הסמן בדף המבוקש ואת התכונה שבה הוא מוצג. לכל תו במתאם המסך מוקצים שני בתים: אחד עבור ערך התו ואחד עבור התכונה.
BH = מספר הדף המבוקש בפקודה.

בחזרה מפסיקה:

AL = ערך ASCII של התו שנמצא במקום שהסמן מצביע עליו.

AH = התכונה המלווה את התו.

9 - כתיבה של התו והתכונה שבה יוצג, במקום שבו נמצא הסמן בדף המבוקש

בקריאה לפסיקה:

AH = 9 כתוב את התו הנמצא באוגר AL.

AL = ערך ASCII של התו שיש להציג.

BL = תכונת התו שיש להציג.

BH = מספר הדף שבו ייכתב התו.

CX = מספר הפעמים שיש לחזור על פעולת הכתיבה.

בחזרה מפסיקה:

בגמר הכתיבה הסמן יהיה לאחר התו האחרון שנכתב. אוגר CX שימושי כאשר רוצים לכתוב תו זהה מספר פעמים. במקרה זה קוראים לתכנית פעם אחת ומשנים את ערכו של אוגר CX בהתאם. כתיבה יחידה חוסכת זמן גישה לתכנית BIOS ומאיצה את ביצוע התכנית.

0A - כתיבת תו במקום שבו נמצא הסמן בדף מבוקש

בקריאה לפסיקה:

AH = 0AHex כתוב את התו במקום שבו נמצא הסמן מבלי לשנות את התכונה השייכת לתו המוצג.

AL = ערך התו שיש לכתוב למסך.

BH = מספר דף בו יש לכתוב.

CX = כמה תווים יש לכתוב.

בחזרה מפסיקה:

בסיום הכתיבה, הסמן יוצב לאחר התו האחרון שנכתב. אוגר CX שימושי כאשר רוצים לכתוב תו זהה מספר פעמים. במקרה זה קוראים לתכנית השירות פעם אחת בלבד ומשנים את אוגר CX בהתאם. כתיבה יחידה חוסכת זמן גישה לתכנית BIOS ומאיצה את ביצוע התכנית.

13 - כתיבת מחרוזת למסך

בקריאה לפסיקה:

AH = 13Hex כתוב מחרוזת (String) למסך. הפקודה שימושית כאשר רוצים לכתוב למסך הודעה או סדרה של תווים שונים זה מזה.

AL = 0, 1 המחרוזת מורכבת מתווים בלבד, התכונה שלהם רשומה באוגר BL. אם ערכו של אוגר AL הוא 1, בגמר כתיבה הסמן יעמוד לאחר התו האחרון של המחרוזת. אם ערכו של אוגר AL הוא 0, הסמן לא יזוז ממקומו.

2, 3 המחרוזת מורכבת מאוסף של תווים ותכונות: בית ראשון מייצג את ערך התו ובית שני מייצג התכונה שלו, בית שלישי מייצג ערך תו שני ובית רביעי את התכונה שלו וכו'. כל תו

יירשם על גבי המסך על פי התכונה המלווה אותו. אם AL=2, הסמן יישאר במקומו ולא יזוז כתוצאה מהפקודה. אם AL=3 הסמן יוצב לאחר התו האחרון שנכתב.
משמעות התווים CR, LF, TAB היא כתווי בקרה למסך. תכנית BIOS תתייחס אליהם כתווי בקרה, תבצע את הפקודה המיוצגת על ידם, ולא תציג אותם על המסך: הערך CR יורה לסמן לחזור לתחילת שורה, הערך LF יגרום שהסמן ירד שורה והערך TAB יגרום לתזוזת הסמן לנקודת העצירה הבאה במסך (לפי כיוון השפה).

BH = מספר הדף שבו יש לכתוב את המחרוזת.
CX = אורך המחרוזת.
DX = מספר שורה ועמודה לתחילת כתיבה.
ES = כתובת המקטע (Segment Address) שבו נמצאת המחרוזת.
BP = כתובת ההיסט (Offset) של המחרוזת במקטע.

פונקציות לטיפול בגרפיקה של פסיקת 10Hex

0B - קביעת צבע משטח העבודה (Pallete)

בקריאה לפסיקה:

AH = 0BHex קבע את צבע משטח העבודה על גבי המסך. משטח העבודה קובע את הרכב הצבעים המופיעים על גבי המסך מכיון שכל צבע הנכתב על המסך יעורבב עם צבע המשטח והצבע המתקבל יהיה שילוב של צבע הנקודה וצבע המשטח.
BH = 0 כיוון צבע הרקע וגבולות משטח העבודה.
1 כיוון צבע המשטח.
BL = מספר הצבע שבו יש להשתמש למשטח, או לגבולות.

0C - סימון נקודה (Pixel)

בקריאה לפסיקה:

AH = 0CHex כתיבת נקודה במקום המבוקש על גבי המסך ובצבע המבוקש.
AL = 0-127 הצבע בו תופיע הנקודה. כתיבה מחדש של הנקודה תמחק את הנקודה הנמצאת במקום שבו יש לכתוב את הנקודה החדשה.
128-256 הנקודה תיכתב במקום המבוקש על ידי ביצוע פעולת XOR בין צבע הנקודה החדשה לבין צבע הנקודה המסומנת באותו מקום על המסך. פעולת XOR שימושית כאשר רוצים לצייר דמות על המסך ולהניע אותה, מבלי למחוק את הציור שנמצא מתחתיה.

לדוגמה:

צבע נקודה קודמת	צבע נקודה חדשה	צבע הנקודה לאחר כתיבה
0	0	0
0	1	1
1	0	1
1	1	0

CX = מספר העמודה על גבי המסך (קואורדינטה X).
DX = מספר השורה על גבי המסך (קואורדינטה Y).

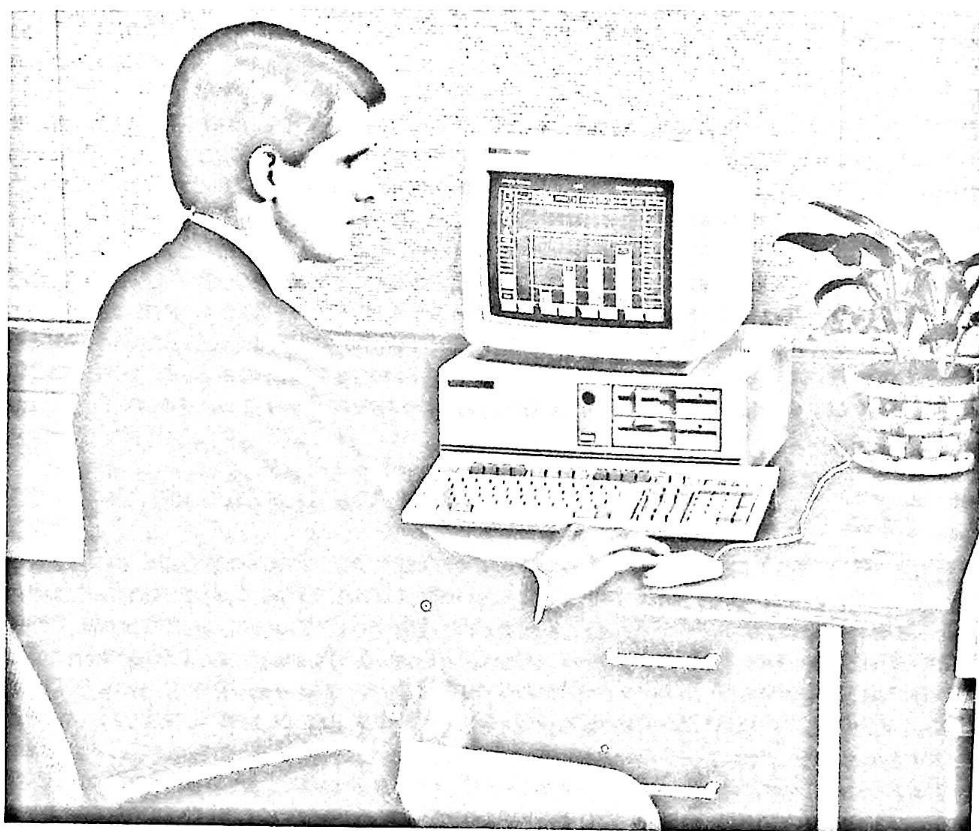
0D - החזרת צבע נקודה במסך

בקריאה לפסיקה:

AH = 0DHex קרא את צבע הנקודה מהמקום המבוקש על גבי המסך.
CX = מספר עמודה במסך (קואורדינטה X).
DX = מספר השורה במסך (קואורדינטה Y).
BH = מספר הדף.

בחזרה מפסיקה:

AL = צבע הנקודה המבוקשת.



4.2 המקלדת במחשב

המקלדת (לוח מקשים - Keyboard) הינה הממשק המרכזי בין המשתמש לבין המחשב האישי. היא מאפשרת למשתמש להתקשר אל המחשב כדי לתת לו הוראות וכדי להריץ בו תכניות שונות. הממשק בין המקלדת לבין המחשב נראה פשוט ומובן מאליו, אך התהליך שבין לחיצה על מקש כלשהו בלוח לבין הצגתו של התו התואם על המסך אינו פשוט. המקלדת מתחברת אל הלוח הראשי של המחשב באמצעות פתיל המחובר בדרך כלל מצידו האחורי. פתיל זה מזין את המקלדת מתח הדרוש לפעולתה ומאפשר את התקשורת בינה לבין המחשב.

הנתונים הנשלחים מהמקלדת אל המחשב, גורמים להפעלת מנגנון פסיקה: פסיקת חומרה Irq1 בבקר הפסיקות; פסיקה 9Hex במערכת המחשב. מנגנון הפסיקה מבקש מהמחשב, כמו בכל פסיקה אחרת, להפסיק את פעולתו לצורך טיפול באירוע חריג, במקרה זה - קבלת הנתונים מהמקלדת. הנתונים המתקבלים הם על מצב המקשים - איזה מקש נלחץ ואיזה מקש שוחרר. התקשורת בין המקלדת לבין המחשב היא תקשורת טורית. כלומר, הנתונים זורמים למחשב סיבית אחר סיבית, 8 סיביות לכל תו מוקש. בסיום ההקשה של כל תו משודרת למחשב סיבית לגילוי שגיאות (Parity), המאפשרת לו לבדוק אם הנתון שנקלט הינו תקין (ראה הסברים על מנגנון Parity בהמשך, בסעיף תקשורת טורית).

במקלדת קיים מעבד נוסף השולט על פעולתה ומאפשר למחשב להתקשר אליה. המחשב יכול לבקש ממעבד המקלדת שירותים שונים, כגון שליטה על מהירות הדפסה (במחשבי AT ומעלה בלבד), **אתחול** (Reset) של המקלדת, **אפשור/חסימה** (Enable/Disable) של המקשים, שליחה מחדש של תו שהתקבל שגוי ועוד.

בתהליך BOOT מבקש המחשב מהמקלדת לבדוק את עצמה ולהודיע אם מתגלית תקלה כלשהי. בשלב הבדיקה, אם מקש כלשהו לחוץ מסיבה של תקלה או במכוון, הדבר יגרום להודעת שגיאה, כי המשמעות הקבועה היא "תקלה". כל פקודה הנשלחת מהמחשב למקלדת מחייבת אותה לשלוח אישור קבלה בתוך פרק זמן נתון. חוסר תגובה של המקלדת לפקודה שנשלחה אליה מזוהית כתקלה.

4.2.1 ההקשה וקליטת ערך המקש

המעבד של המקלדת סורק כל הזמן את הלוח כדי לבדוק אם נלחץ מקש כלשהו. המעבד מזהה מקש שנלחץ ושולח למחשב מידע על כך. הצירוף הבינארי הנשלח מהמקלדת אינו הצירוף המתאים לתו על פי טבלת American Standart) ASCII Code for Information Interchange). מידע זה הוא **קוד סריקה** (Scan Code), אשר מציין את מיקומו של המקש שנלחץ במערך המקשים על פני הלוח. כלומר, המקשים ממוספרים על פי מיקומם ולא על פי הצירוף המתאים להם בטבלת ASCII.

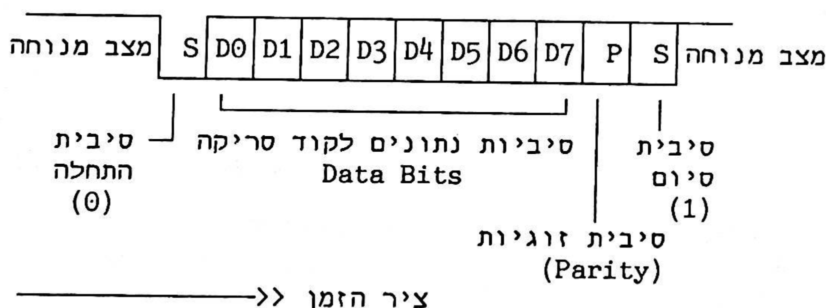
לחיצה על מקש גורמת לשליחת הודעה למחשב כי המקש נלחץ, ושחרורו יגרום לשליחת קוד הסריקה עם הודעה שהוא שוחרר. במחשבי AT ומעלה מבחין המחשב בקוד השחרור על ידי זיהוי קידומת מתאימה לפני קוד הסריקה של המקש שפירושה: "שוחרר המקש שקוד הסריקה שלו מוצג כאן". במחשבי XT, המספרים בתחום 0-79Hex הם קודי סריקה של מקש שנלחץ. תוספת של 80Hex לקוד מסמן

מקש משוחרר. לפיכך, הצירוף 01Hex מודיע למחשב שנלחץ מקש שקוד הסריקה שלו 1, והצירוף 81Hex מודיע למחשב כי המקש שקוד הסריקה שלו 1 שוחרר.

כאשר מקש לחוץ באופן רצוף, ממשיך לוח המקשים לשלוח את קוד הסריקה שלו למחשב באופן רצוף. הזמן שבין הלחיצה הרצופה על המקש לבין התחלת שליחת קודים רציפים של אותו מקש למחשב, נקרא **זמן השהיה** (Delay Time). הזמן החולף בין קוד רצוף אחד לשני, נקרא **קצב הקלדה** (Typematic Rate). שני פרמטרים אלה קבועים מראש במחשבי XT. במערכות AT ומעלה ניתן לשלוט על שני פרמטרים אלה בעזרת פקודת MODE בגירסה 4.0 ומעלה של מערכת ההפעלה DOS. כאשר שני מקשים נלחצים בו זמנית, המקש האחרון שנלחץ הוא זה אשר ישודר בצורה רציפה למחשב (רק אם השארנו אותו לחוץ באופן קבוע).

4.2.2 התקשורת בין המקלדת לבין המחשב

המקלדת משדרת למחשב את קוד הסריקה (Scan Code) בצורה טורית, **כמסגרת מידע**, כפי שמוסבר בסעיף תקשורת אסינכרונית. אם המחשב אינו מגיב לבקשת השירות של המקלדת, היא שומרת את זיהוי המקשים שנלחצו עד אשר המחשב יתן אישור לשליחתם. במקלדת ישנו זיכרון המאפשר לשמור עד 14 לחיצות מקשים. כאשר המחשב קולט את המקשים הוא מעביר נתונים אלה אל זיכרון RAM השמור למטרה זו.



שידור מסגרת מידע מהמקלדת למחשב

4.2.3 זיהוי של המקש בתכנית ROM BIOS

תהליך הקליטה של קוד הסריקה מהמקלדת מורכב מבקשת שירות של המקלדת ותגובה לבקשה זו. בקשת השירות מהמקלדת יוצרת במחשב פסיקת חומרה מספר 9.

המחשב מקצה בזיכרון RAM מקום עבור נתוני המקלדת שבו אפשר לשמור בתור עד 14 קודי סריקה (תווים). הקשה של למעלה מ-14 מקשים לפני שמערכת ההפעלה סיימה לבצע תכנית קודמת, תגרום להשמעת צפצוף של המחשב המציין כי המקום המוקצה מלא ואין באפשרותו לקבל מקשים נוספים.

כאשר הנתון נקלט מלוח המקשים, תכנית ROM BIOS המטפלת בפסיקת המקלדת מתחילה להתבצע ומבררת איזה מקש נלחץ ורושמת זאת בזיכרון RAM, במקום

שהוקצה לנתוני המקלדת. היא מתרגמת את ערך קוד הסריקה שמתקבל לקוד ASCII של המקש, כי קוד הסריקה אינו מובן כתו. נתאר עתה את השלבים:

בשלב ראשון, תכנית BIOS מבררת אם המקש שנלחץ הוא אחד המקשים בעלי תפקיד או משמעות מיוחדים. כידוע, מקשי Shift, Ctrl, Alt משנים את משמעות המקש אשר נלחץ יחד איתם. התכנית מסמנת לעצמה את מצב המקשים האלה (ראה טבלה בנספח). בשלב הבא מזהה תכנית BIOS אם המקש הנלחץ הוא אחד ממקשי הבקרה המשניים, שיש להם פעולה נגדית בכל לחיצה עליהם: מקשי Ins, Num Lock, Scroll Lock, Caps. היא מסמנת את מצבם במקום מיוחד בזיכרון RAM.

בשלב הבא מנסה תכנית BIOS לתת משמעות למקש הנלחץ על ידי תרגום של קוד הסריקה שלו לקוד ASCII. קוד ASCII וקוד הסריקה נשמרים בזיכרון RAM של המחשב האישי במקום המיועד למקלדת.

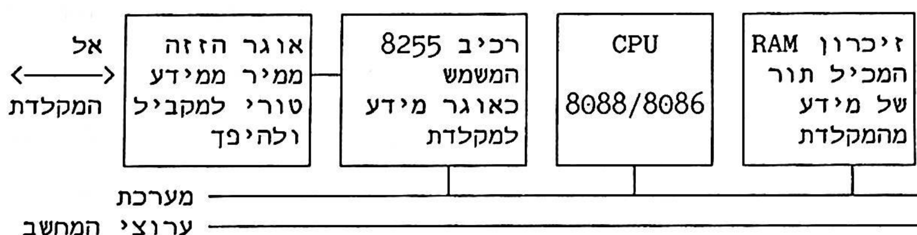
אפשרות גישה למקלדת ניתנת למתכנת על ידי תכנית ROM BIOS באמצעות פסיקת תוכנה מספר 16Hex. פסיקת שירות זו מאפשרת לברר את מצב מקשי הבקרה השונים, לברר אם מקש כלשהו נלחץ וערכו נשמר בתור וגם לקרוא מקש מהתור (ראה תכנית דוגמה לבדיקת מצב מקשי בקרה בנספח תכניות הדוגמה). שירותי פסיקת השירות מתוארים בהמשך.

בלחיצה על מקש פונקציה במקלדת, כמו F1-F12, Home, Page Up וכו', תחזיר תכנית ROM BIOS את הערך 0 בטבלת ASCII כקוד ASCII של המקש שנלחץ ואת קוד הסריקה שלו. תכנית השירות של מערכת ההפעלה (או שפות התכנות השונות) מתרגמת את קוד הסריקה למקש פונקציה מתאים ומאפשרת לתכנית לזהות אותו (תכנית דוגמה ניתן למצוא בנספח תכניות הדוגמה).

4.2.4 מקלדת XT ומקלדת AT ומעלה

במחשב PC המבוסס על מעבד 8088 הגישה אל המקלדת נעשית באמצעות כתיבת נתון לרכיב הכניסות - Port (8255) המחובר לאוגר הזזה טורי. אוגר זה שולח את הנתון למקלדת בצורה טורית. במחשבי AT החיבור מתוחכם יותר ונעשה דרך מיקרו בקר (מעבד 8042) הנמצא על הלוח הראשי במחשב ואחראי על התקשורת הטורית למקלדת וממנה. חיבור המקלדת דרך הבקר גמיש יותר מהחיבור במחשבי XT, כי הוא מאפשר להרחיב את אפשרויות השימוש במקלדת. שינויים בתקשורת למקלדת נעשים במקרה זה בתוכנה, ללא שינויי חומרה כלשהם במערכת.

חיבורי החומרה בין המקלדת למחשב זהים במחשבים השונים (מלבד PS/2) ולכן מבנה הפתיל והמחבר לקישור המקלדת למחשב זהים. על מנת להסב את המקלדת מצורת פעולה המתאימה למחשבי XT לצורת פעולה המתאימה למחשבי AT, קיים בתחתיתה מתג, שבאמצעותו ניתן לשנות את צורת הפעולה (רק בלוחות מקשים המאפשרים זאת). ממשק התוכנה של מחשבי AT מורכב יותר ויכול לשלוט על זמן ההשהיה ועל קצב ההקשה של התווים (Delay Time ו-Typematic Rate בהתאמה).



התקשורת למקלדת במחשבי PC



התקשורת למקלדת במחשבי AT

4.2.5 משמעות התווים במחשב

משמעות התווים המועברים בין המקלדת לבין המחשב חורגת מהתחום הצר של תקשורת בין שתי יחידות אלו. נסביר זאת עתה בהרחבה ואת השימוש נלמד בהמשך הפרק.

קובץ, תכנית או אוסף נתונים כלשהו מורכב מתווים. כל קובץ מורכב מנתונים ברוחב התלוי באופי הנתונים השמורים בו. לדוגמה, בקובץ המכיל נתונים מסוג **מלה** (Word) כל נתון הוא ברוחב של 16 סיביות. בקובץ המכיל תכנית בשפת מכונה כל נתון הוא פקודה למעבד, אשר הרוחב שלה משתנה בין 8 ל-64 סיביות לפי סוג הפקודה. בקובץ המכיל טקסט, כל נתון בקובץ הוא ברוחב 8 סיביות.

8 סיביות הוא מספר "קסם". סוגים רבים של נתונים הם ברוחב 8 סיביות והם נפוצים במערכות תקשורת רבות, הן מקביליות והן טוריות. לנתון ניתן להתייחס במספר דרכים, על פי ההגדרה בתקנים:

* **בית** (Byte): ערך מספרי חיובי שלם בתחום 0 עד 255.
* **מספר שלם קצר** (Short Integer): ערך מספרי שלם 127- עד 128.
* **תו טקסט** (Character): התייחסות על פי הערכים בטבלת ASCII.

שני הסוגים הראשונים הם מספרים פשוטים, בינאריים. הטקסט מורכב יותר ומבוסס על תקן ASCII שהוא התקן הנפוץ ביותר כיום. מן הראוי לציין שישנם תקנים אחרים, בעיקר למחשבים גדולים (למשל תקן EBCDIC).

בהתייחסות לתווי טקסט, נחלק את טבלת ASCII לשלושה תחומים:

* 0-31 ערכים עבור תווי בקרה.
* 32-127 ערכים עבור תווי טקסט רגילים.
* 127-255 ערכים עבור תווי טקסט עליונים.

קבוצת התווים העליונים מתחלקת על פי הפירוט הבא:
* 128-168 תווים עבור שפה שנייה (עברית, ערבית וכו').
* 169-255 תווים גרפיים (עבור טבלאות וכו').

בתוך קובץ יכולים להופיע תווים בכל ערך שהוא. בקובצי טקסט קיימת משמעות מיוחדת לתווי הבקרה, אשר שלושה מהם מובנים על ידי מרבית ההתקנים:
* ערך 10 מסומן כ-NL או LF, אשר מודיע להתקן להתקדם שורה.
* ערך 13 מסומן כ-CR ומודיע להתקן לחזור לתחילת השורה.
* ערך 26 מסומן כ-EOF ומודיע להתקן שזהו סוף טקסט.

כאשר רוצים להסתיר ממשתמש חלק של קובץ טקסט, ניתן להכניס את הערך 26 במקום שממנו רוצים להסתיר. תכניות תקניות להצגת טקסט או לעריכת קובצי טקסט, יציגו את הקובץ רק עד למקום בו קיים ערך זה, ולא יציגו את המשכו.

התקנים שונים מתייחסים לתווי בקרה הנשלחים אליהם באופן שונה. כך למשל,

* תו שערכו 13 (CR)
  מדפסת - החזרת הראש הכותב לתחילת השורה.
  מסך דרך BIOS - החזרת הסמן לתחילת השורה במסך.
  מסך בכתיבה ישירה - נכתב תו שצורתו תו מוסיקלי.

* תו שערכו 7 (Bell)
  מדפסת - לא תבצע דבר.
  מסך דרך BIOS - נשמע צפצוף קצר של הרמקול במחשב.
  מסך בכתיבה ישירה - נכתב תו שצורתו עיגול מלא.

כאשר משדרים קבצים דרך רשתות תקשורת, קיימת בעיה הנובעת מכך שאין תקן אחיד. בחלק מהרשתות יש **שקיפות** (Transparency) מלאה לתווים המועברים דרכן, כלומר כל תו שנשלח מצד אחד מתקבל בצד השני ללא שינויים מבלי להתייחס לתוכנו. לעומת זאת, קיימות רשתות שאינן מעבירות תווים עליונים, או שהן משתמשות בתווי הבקרה. כלומר, רשתות אלו אינן שקופות לנתונים ושימוש בתווי בקרה או תווים עליונים עלול לשבש את התקשורת.

בנספח מוצגות טבלאות ASCII תקני, טבלאות ASCII הכוללות שפה שנייה עברית וטבלה של תווי בקרה. כאשר כותבים תכניות תקשורת או ממשק להתקנים שונים רצוי להכיר את תווי הבקרה, או לפחות את החשובים שבהם.

תכנית BIOS מאפשרת לשלוח צירופים מיוחדים מהמקלדת למחשב במספר דרכים המקובלות גם במחשבים שאינם תואמי יבמ.

4.2.6 צירופים מיוחדים של מקשים

התכנית המטפלת בקליטת נתונים מהמקלדת מטפלת בצורה שונה בצירופי מקשים בעלי משמעות מיוחדת. צירופים אלה אינם ניתנים לקריאה כקלט רגיל על ידי התכנית.

לחיצה על מקש Ctrl ואחר כך על אחת מאותיות האנגלית A-Z:

Ctrl + A תחזיר את הערך 1.
Ctrl + B תחזיר את הערך 2.
Ctrl + C תחזיר את הערך 3.
עד
Ctrl + Z תחזיר את הערך 26.

* לחיצה רגילה על Enter תחזיר את הערך 13 (CR).
* לחיצה על Ctrl + Enter תחזיר את הערך 10 (LF).

צירוף המקשים Alt+Ctrl+Del

הצירוף Alt+Ctrl+Del הוא בעל משמעות מיוחדת. כאשר תכנית הטיפול במקלדת מזהה צירוף זה, היא גורמת לתהליך BOOT של המחשב: איפוס בתוכנה של מערכת המחשב, ולא Reset בחומרה. באתחול תוכנה של המערכת לא מבוצעים כל שלבי בדיקת החומרה, אלא רק שלבי טעינה של מערכת ההפעלה DOS. פרק הזמן הנדרש לטעינת מערכת ההפעלה בצורה זו מהיר יותר מתהליך BOOT בחומרה.

צירוף מקשים זה מושג על ידי לחיצה על מקש Alt, וכשהוא לחוץ - גם לחיצה על מקש Ctrl, וכאשר שניהם לחוצים, הוספה של לחיצה על Del. כך שבמצב הסופי נקבל Alt+Ctrl+Del לחוצים ביחד. **סדר הלחיצה**, ברוב המחשבים, חייב להישמר בדיוק כך, אחרת הצירוף נחשב לצירוף אחר.

צירוף המקשים Alt ומקשי Numeric Keypad

במצב זה מאפשרת תכנית BIOS למשתמש להכניס את הצירופים השונים של טבלת ASCII על פי ערכם המספרי. הקשה רגילה על המקלדת אינה מאפשרת להכניס כקלט את כל צירופי טבלת ASCII, אלא רק את מקצתם. לדוגמה, הקשה על Alt ועל הספרות הבונות את המספר 219 תשלח למחשב את הסימן שמספרו בטבלת ASCII הוא 219. סימן זה הוא ריבוע מלא.

צירוף מקשים זה מושג על ידי לחיצה על מקש Alt, וכשמקש זה לחוץ הוספה של לחיצה על מקש כלשהו **מלוח המקשים המספרי הנמצא בצד ימין** של המקלדת. לאחר סיום הקשת המספר יש לשחרר את מקש Alt, כדי שהתו יישלח למחשב.

צירוף המקשים Shift+PrtSc, או מקש Print Screen

צירוף מקשים זה מפורש על ידי התכנית הקוראת מהמקלדת כבקשה להדפסת מסך. התכנית תבצע פסיקת תוכנה מספר 5, אשר תדפיס את תוכן המסך על המדפסת (Hard Copy של המסך).

צירוף מקשים זה מושג על ידי לחיצה על מקש Shift וכשהוא לחוץ, הוספה של לחיצה על מקש PrtSc. יש מקלדות שבהן אין צורך ללחוץ על מקש Shift, אלא על מקש Print Screen בלבד.

כאשר המחשב נמצא במצב גרפי ומעוניינים להדפיס את המסך בצורה גרפית, לא נוכל לבצע זאת בעזרת תכנית שירות מספר 5 של BIOS. בכדי להדפיס מסך גרפי, יש לבצע תחילה את התכנית Graphics של מערכת ההפעלה (פקודה חיצונית). תכנית זו נטענת כתכנית TSR, מחליפה את פסיקת שירות מספר 5 של BIOS ומאפשרת הדפסת מסכים גרפיים.

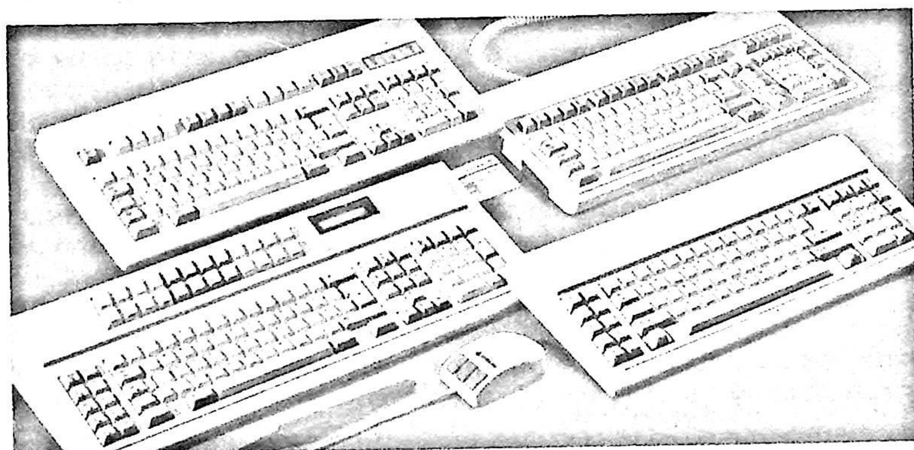
תכנית Graphics מגירסה 4 של DOS מאפשרת להדפיס גם מסכים גרפים מתקדמים יותר כמו מסכי EGA ומעלה. בגירסאות נמוכות יותר של מערכת ההפעלה DOS לא ניתן להדפיס תוכן מסך גרפי EGA ומעלה באמצעות לחיצה על Shift+PtrSc.

צירוף המקשים Ctrl+Break

צירוף מקשים זה גורם לפסיקה בפעולת התכנית. הצירוף מושג על ידי לחיצה על מקש Ctrl וכשהוא לחוץ, הוספה של לחיצה על מקש Break. כאשר נלחץ צירוף מקשים Ctrl+Break, תוכנת BIOS של לוח המקשים קוראת לפסיקת שירות מספר 1BHex.

תכנית יכולה לחסום את הפסיקה הזו, לבצע שגרה מיוחדת לטיפול בצירוף מקשים Ctrl+Break, או **לאפשר** (Enable) פסיקה. תכנית המאפשרת את הפסיקה הזו ומזהה צירוף זה, תסיים את פעולתה באמצע ותחזור למערכת ההפעלה.

ניתן להגדיר שגרה מיוחדת שתתבצע כאשר המשתמש יקיש Ctrl+Break. לשם כך צריך לעדכן בווקטור פסיקת 1BHex את כתובת השגרה המטפלת בצירוף המקשים. השגרה יכולה לשמור מידע על מצב המערכת והאוגרים לפני חזרה למערכת ההפעלה. לחיצה על צירוף זה וטיפול לא נכון בפסיקה, עלולים לגרום לאובדן מידע של התכנית.



4.2.7 שימוש בפסיקות BIOS לצורך עבודה

4.2.7.1 פקודות לכל סוגי המקלדות — פסיקת BIOS 16Hex

0) קריאת תו מהמקלדת

בקריאה לפסיקה:

AH = 0 החזר את התו הנמצא בראש התור במקלדת לאוגר AL.
החזר את קוד הסריקה של המקש לאוגר AH.

בחזרה מהפסיקה:

AL = ערך ASCII של המקש יימצא באוגר זה, או הערך 0 אם המקש הוא מקש פונקציה.

AH = קוד הסריקה (Scan Code) של המקש.

1) בירור מצב התור במקלדת (האם נלחץ מקש כלשהו?)

בקריאה לפסיקה:

AH = 1 בדוק את מצב התור במקלדת והחזר 0 בדגל האפס (Zero), אם מקש כלשהו מחכה בתור. הדבר שימושי במשחקים למיניהם. התכנית ממשיכה בביצוע עד שיילחץ מקש כלשהו.

בחזרה מפסיקה:

Flags = אם תוכן דגל האפס הוא 0, ישנו מקש בתור של המקלדת המחכה לקריאה מהמקלדת. אם תוכן דגל האפס שונה מ-0, אין מקש המחכה בתור.

AL, AH = כמו הפונקציה 0.

2) בירור מצב מקשי הבקרה במקלדת

בקריאה לפסיקה:

AH = 2 החזר את מצב מקשי הבקרה המתאימים באוגר AL.

בחזרה מפסיקה:

AL = מכיל את מצב מקשי הבקרה על פי הפרוט הבא:

סיבית	משמעות (1 משמעותו מקש לחוץ)
0	מקש Shift ימני
1	מקש Shift שמאלי
2	מקש Ctrl (ימני או שמאלי)
3	מקש Alt (ימני או שמאלי)
4	מקש Scroll Lock
5	מקש Num Lock
6	מקש Caps Lock
7	מקש Ins

(תכנית דוגמה לטיפול במקשי בקרה נמצאת בנספח)

4.2.7.2 פקודות למקלדות מורחבות בלבד — פסיקת 16Hex BIOS

הפקודות הבאות תפעלנה רק **בלוח מקשים מורחב** (Enhanced Keyboard) הנמצא במערכות AT ומעלה ולא בלוח מקשים רגיל. בהמשך מוסבר תהליך הזיהוי אם זוהי מקלדת מורחבת או לא.

<u>3) קבע קצב עבודה של המקלדת</u>

בקריאה לפסיקה:

AH = 3 קבע זמן השהיה (Delay) עד לזיהוי לחיצה רצופה על מקש.
קבע קצב הקשה (Rate), או זמן המתנה בין מקש למקש בשידור רצוף.

AL = 5 קבע השהיה וקצב שידור.

BH = 0 המתנה של 0.25 שנייה עד לזיהוי לחיצה רצופה.
1 המתנה של 0.50 שנייה עד לזיהוי לחיצה רצופה.
2 המתנה של 0.75 שנייה עד לזיהוי לחיצה רצופה.
3 המתנה של 1.00 שנייה עד לזיהוי לחיצה רצופה.

BL = 0-1F זמן המתנה בין שליחת מצירוף אחד לשני בלחיצה רצופה.
ערך 0 יתן את הקצב הגבוה ביותר,
ערך 1FHex יתן את הקצב הנמוך ביותר.

<u>5) כתוב ערך מקש למקלדת</u>

בקריאה לפסיקה:

AH = 5 כתוב ערך למקלדת. במקלדת ישנו תור המכיל 16 כניסות למקשים שנלחצו. ניתן באמצעות פסיקה זאת לשלוח מקשים לתכנית.

CL = ערך התו לפי ASCII.

CH = קוד הסריקה של המקש.

בחזרה מפסיקה:

AL = 1 התור במקלדת מלא.
<> 1 התו נכתב בהצלחה למקלדת.

<u>10) קריאת תו מהמקלדת</u>

בקריאה לפסיקה:

AH = 10Hex החזר את התו הנמצא בראש התור במקלדת לאוגר AL,
החזר את קוד הסריקה של המקש לאוגר AH.

בחזרה מפסיקה:

AL = ערך ASCII של המקש יימצא בתוך האוגר או הערך 0 אם המקש הוא מקש פונקציה.

AH = קוד הסריקה של המקש.

ההבדל בין תכנית שירות זו לבין תכנית שירות 0 הוא ביכולת המקלדת להבחין בין מקשי Alt, Ctrl ו-Shift הימני והשמאלי.

11) בירור מצב התור במקלדת (האם נלחץ מקש כלשהו?)

בקריאה לפסיקה:

AH = 11Hex בדוק את מצב התור במקלדת והחזר 0 בדגל האפס (Zero) אם יש מקש כלשהו המחכה בתור. שימושי במשחקים למיניהם. המשך בביצוע התכנית עד שיילחץ מקש.

בחזרה מפסיקה:

Flags = תוכן דגל האפס 0, ישנו מקש בתור המקלדת המחכה לקריאה. אם תוכן דגל האפס שונה מ-0, אין מקש המחכה בתור.

AH,AL = כמו בפונקציה 0

ההבדל בין תכנית שירות זו לבין תכנית שירות 1 הוא ביכולת המקלדת להבחין בין מקשי Alt, Ctrl ו-Shift הימני והשמאלי.

12) בירור מצב מקשי בקרה במקלדת

בקריאה לפסיקה:

AH = 12Hex החזר את מצב מקשי הבקרה המתאימים באוגרים AL ו-AH.

בחזרה מפסיקה:

AL = מצב מקשי הבקרה על פי הפירוט הבא:

סיבית	משמעות (1 משמעותו מקש לחוץ).
0	מקש Shift ימני.
1	מקש Shift שמאלי.
2	מקש Ctrl (ימני או שמאלי).
3	מקש Alt (ימני או שמאלי).
4	מקש Scroll Lock מאופשר (Enabled).
5	מקש Num Lock מאופשר.
6	מקש Caps Lock מאופשר.
7	מקש Ins השתנה מאז הפעם הקודמת.

AH = ערכי מקשי הבקרה הבאים: (1 משמעותו מקש לחוץ)

0	מקש Ctrl שמאלי.
1	מקש Alt שמאלי.
2	מקש Ctrl ימני.
3	מקש Alt ימני.
4	מקש Scroll Lock.
5	מקש Num Lock.
6	מקש Caps Lock.
7	מקש SysReq.

4.2.7.3 זיהוי סוג המקלדת

על מנת לזהות איזה סוג מקלדת קשור למחשב, צריך לפעול כך:

- השתמש בפונקציה 5 בכדי לרשום ללוח המקשים.
  כתוב למקלדת את הערכים AA Hex כערך מקש, 55 Hex כקוד סריקה.
  כתוב למקלדת את הערכים 55 Hex כערך מקש, AA Hex כקוד סריקה.

- השתמש בפונקציה 10Hex בפסיקה 16Hex בכדי לקרוא מקשים אלה מהמקלדת.
  אם קראת את שני המקשים על פי הסדר שנכתבו, לפניך מקלדת מורחבת,
  ואם לא - המקלדת הינה רגילה.

4.2.8 פקודות מערכת ההפעלה לטיפול במקלדת

4.2.8.1 יצירת קובץ ישירות מהמקלדת

כדי ליצור קובץ ולשמור אותו על גבי דיסק/דיסקט עלינו להשתמש בתכנית עריכה כלשהי. רצוי שגם תוכל לערוך בו שינויים. כאשר לא מצויה תכנית כזאת, אפשר ליצור קבצים קטנים (כמו AUTOEXEC.BAT או CONFIG.SYS) על ידי שימוש במקלדת ובפקודה COPY. הפקודה COPY מאפשרת ליצור קבצים בלבד ולא מאפשרת לערוך בהם שינויים או תיקונים.

תבנית הפקודה: `COPY CON File-Name`

כאשר:
COPY - פקודה להעתקת קבצים.
CON - שמה של המקלדת במערכת ההפעלה.
File-Name - שם הקובץ שיש ליצור.

לאחר רישום הפקודה יש להקיש את תוכן הקובץ. כל שורה הינה רשומה נפרדת. נסיים על ידי לחיצה על Ctrl+Z (^Z – סוף נתונים) ו-Enter.

דוגמה:
הפקודה ורצף השורות שאחריה
(תשובות מערכת ההפעלה מודגשות)

```
A> COPY CON CONFIG.SYS
FILES=30
BUFFERS=30
^Z
1 Files Copied
A>
```

כתוצאה נקבל בכונן A קובץ המגדיר את תצורת המערכת ומכיל שתי פקודות (FILES ו-BUFFERS).

4.2.8.2 בקרת מהירות המקלדת (מערכות AT ומעלה)

השליטה על המקלדת אפשרית בגירסה 4 ומעלה של מערכת ההפעלה DOS באמצעות הפקודה: MODE.
קודה זו מאפשרת שליטה על קצב ההקשה ושידור ערכי המקשים למחשב (Typematic Rate) ועל פרק הזמן לזיהוי השליחה הרצופה (Delay Time).

תבנית הפקודה: MODE CON: RATE=rate DELAY=delay

כאשר:

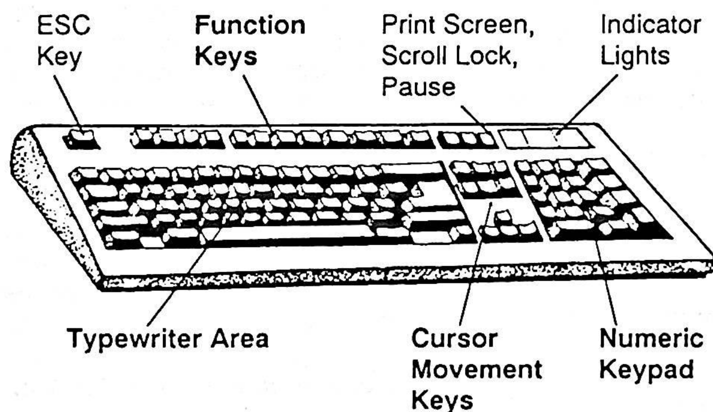
rate - פרמטר זה קובע את קצב השליחה למחשב של קוד הסריקה של מקש הלחוץ באופן קבוע (Typematic Rate).
יש לקבוע ערך בין 1 ל-32. ככל שהמספר גבוה יותר, קצב השליחה של קוד הסריקה גבוה יותר.

delay - פרמטר זה קובע את זמן ההשהיה של המקלדת עד לזיהוי של לחיצה רצופה על המקש (Delay Time). חייבת להיות השהיה כלשהי, מכיון שאחרת, כל לחיצה לא זהירה על כל מקש שהוא תגרום להקשה כפולה, או יותר.
ערכים אפשריים הם בין 1 ל-4, אשר מציינים זמן השהיה של 0.25, 0.5, 0.75, 1 שנייה (4 מציין שנייה ו-1 מציין 0.25 שנייה).

לדוגמה, הפקודה MODE CON: RATE=32 DELAY=4

תגרום לכך, שהמקלדת תמתין 1 שנייה, עד שהיא תזהה לחיצה רצופה על מקש כלשהו ואז תשלח את קוד הסריקה שלו למחשב בקצב של 32 קודי סריקה (הקצב הגבוה ביותר).

הערה:
בקרת מהירות המקלדת שימושית מאוד בתכניות המחייבות חזרה במהירות גבוהה על מקשים, כמו תכניות גרפיות, משחקים, מעבדי תמלילים ועוד.



4.3 מדפסת מקבילית

המדפסות הנפוצות במחשבים אישים הן מדפסות מקביליות. המדפסת אינה חלק מהמחשב האישי וחיבורה אליו אינו חיוני לפעולתו התקינה. היא דרושה רק כאשר ברצוננו להוציא פלט מודפס על גבי נייר. פלט זה יכול להיות תווים, מתוך טבלת ASCII או גרפיקה. בדרך כלל נעסוק בהדפסת תווים, וגם אז לא נוכל להדפיס בכל המדפסות את כל התווים שיש להם תבנית גרפית בטבלת ASCII. הדפסה גרפית אפשרית במדפסות מסוימות ובתנאים שנלמד בהמשך.

המדפסת מקבלת את המתח הדרוש לפעולתה באופן ישיר מהרשת. אותות הבקרה והנתונים. להדפסה נשלחים אליה מהכרטיס המתאם המקבילי הנמצא במחשב. את המדפסת ניתן לכוון לאופן הפעולה הרצוי (קצב, צפיפות ועוד) בדרכים אחדות:

א. באמצעות מפסקים (Dip Switches) הנמצאים על המדפסת.
ב. על ידי תכנית Setup של המדפסת.
ג. על ידי שליחת פרמטרים מהמחשב. מרבית הפרמטרים ניתנים לשינוי ומאפשרים הדפסה באופנים שונים (Escape Sequence).

נזכיר שיש גם מדפסות טוריות שמתחברות למחבר הטורי. נדון בכך בהמשך הפרק, בסעיף על מחבר טורי.

4.3.1 סוגי מדפסות

בשוק קיימים מספר סוגים של מדפסות הפועלות בפרוטוקול מקבילי.

מדפסת מניפה

במדפסת מניפה כל התווים נמצאים על גבי דיסקית עגולה (מניפה) או ראש כדורי. הם יצוקים מעופרת או פלסטיק על גבי המשטח הנושא וניתן לראות את צורתם. כל צירוף הנשלח להדפסה גורם לפעולה מכנית של בחירת התו להדפסה מתוך המניפה או הכדור, ומנגנון ההדפסה גורם להקשה של התו על סרט הדיו שנמצא מעל הנייר.

תהליך ההדפסה מזכיר פעולה של מכונת כתיבה. ההבדל הוא בכך, שהנתונים נשלחים למדפסת מהמחשב ולא מהמקלדת של מכונת הכתיבה. מדפסות מניפה מדפיסות תווים בלבד ואינן יכולות להדפיס גרפיקה. כאשר משתמשים במניפה מסוימת אין אפשרות לשנות את גודל האותיות המודפסות והן מאפשרות צורה מוגבלת של הצגת תווים והדגשתם. להדפסת תווים שונים יש להחליף מניפה.

מדפסת מטריצה (Dot Matrix Printer)

מדפסת מטריצה נקראת לעתים גם "מדפסת סיכות", מכיון שההדפסה בה נעשית באמצעות תבנית (מטריצה) של סיכות שנמצאות על ראש ההדפסה. כאשר כותבים בית (byte) נתונים למדפסת, היא יוצרת בתבנית הסיכות את צורת התו על ידי הבלטת הסיכות היוצרות את תבנית התו הזה. הדבר דומה לשיטה של הצגת

תווים במסך. ההדפסה נעשית בעת ההקשה של תבנית הסיכות על הנייר ומכאן מובן שצפיפות הסיכות בתבנית קובעת את איכות ההדפסה. לאחר כתיבת התו ראש ההדפסה מתקדם למקום שבו יש לכתוב את התו הבא. בדרך כלל קיימות שתי צורות הדפסה:

Draft – הדפסה רגילה, או **הדפסת טיוטא:** ראש המדפסת עובר פעם אחת על כל שורה. בתנועה בכיוון הלוך מודפסת שורה אחת ובתנועת ראש ההדפסה חזרה מודפסת השורה הבאה. בהדפסת טיוטא ניתן לראות בבירור את הנקודות המרכיבות את התו ולעתים גם קשה לקרוא אותו.

Near Letter Quality (NLQ) – **הדפסה באיכות מכתב** (או "כמעט"): באופן הדפסה זה המדפסת חוזרת על כל שורה פעמיים ולכן קצב ההדפסה נמוך פי 4 לערך מקצב הדפסה במצב טיוטא. בחזרה של ראש ההדפסה על השורה הוא מוזז מעט בכיוון אנכי ובכיוון אופקי. תזוזה זו גורמת לכך שכל אות נכתבת פעמיים בהסטה מזערית ועל כן איכות ההדפסה עולה באופן משמעותי. בהדפסת NLQ כמעט ואין רואים את הנקודות המרכיבות את האות והכתב קריא יותר.

מדפסות מטריצה מסוגלות להדפיס צורות גרפיות בנוסף לתווים, ניתן לשנות בהן את תכונות הצגת התווים (הדגשות, הגדלת אותיות, חיקוי פקודות של מדפסות אחרות - אמולציה ועוד). מרבית מדפסות המטריצה מדפיסות את השורה רק לאחר שהתכנית סיימה לכתוב אותה למדפסת והמחשב שלח תו סיום המורה לראש ההדפסה לחזור למקומו.

מדפסת התזת דיו (Inc jet printer)

אופן הפעולה דומה מאוד לאופן הפעולה של מדפסת מטריצה. במדפסת התזת דיו אין תנועה מכנית של הקשה על גבי סרט הדיו. הדיו שעל סרט הדיו נדחף אל הנייר על ידי חשמל סטטי שנמצא על הסיכה, ולא באמצעות הקשה מכנית המתבצעת במדפסת מטריצה. מדפסות התזת דיו שקטות יותר בפעולתן ממדפסות מטריצה ומניפה ואיכות ההדפסה שלהן משופרת יותר, אפילו מזו של מדפסת מטריצה. חסרונן של מדפסות אלו נובע מצריכה גבוהה של סרטי דיו שמחירם גבוה.

מדפסת לייזר (Laser printer)

מדפסת הלייזר מדפיסה דפים שלמים בדרך דומה להעתקת מסמכים. מסיבה זו המדפסת שקטה בפעולתה, שלא כמו מדפסות ההקשה לסוגיהן. דף הפלט נשלח בשלימותו מן המחשב ואז הוא מודפס. לעתים הדף הנוכחי מודפס לפני שהמדפסת קלטה את כל התוכן שלו. מצב זה נוצר אם בתוך פרק זמן סביר המחשב לא שלח נתונים נוספים והמדפסת זיהתה מרווח זמן זה כסיום דף. פרק זמן ההמתנה ניתן לתכנות.

מדפסת הלייזר מדפיסה באמצעות קרן לייזר שמאירה מסך, אשר מועתק כמו בצילום מסמכים לדף המודפס. הקרן נעה על גבי המסך בעזרת מראות מצולעות המסתובבות במהירות גבוהה. דף הנייר מונח על גבי התקן פוטו אלקטרי היוצר מתח חשמלי על גבי הנייר בהתאם לצורת התו שיש להדפיס. טונר (Toner), חומר שנראה כאבקת פחם נמשך לנקודות שהוטענו במטען חשמלי וכך הוא יוצר את צורת התו. הדף עובר עם הטונר שעליו ל"תנור" הנמצא בתוך המדפסת, כי דרוש תהליך חימום להדבקת הטונר על הנייר, ואחר כך ממשיך ויוצא מהמדפסת.

מדפסות הלייזר מאפשרות שליטה בתוכנה על כל פרמטר אפשרי בהצגת תווים: הגדלות והדגשות, גופנים שונים הניתנים לטעינה בתוכנה, חיקוי מדפסות ותוויינים, הדפסת בר-קוד, מדבקות, נייר בדיד, כתיבה ברווחים פרופורציונליים הנותנים דף ברור וקריא ועוד.

מדפסות לייזר יקרות יותר ממדפסות מטריצה, מתוחכמות יותר ואחזקתן יקרה יותר יחסית לסוגי מדפסות קודמים.

מדפסת לעבודה מאומצת (Heavy Duty)

בכל סוגי המדפסות שצוינו לעיל ישנם דגמים ביתיים וישנם דגמים לעבודה מאומצת (Heavy Duty). הדגמים הביתיים אינם מיועדים לפעילות הדפסה ממושכת, אם כי במקרים מיוחדים הן יכולות לעמוד בכך לזמן קצר. אחוז הבלאי במקרים אלה גבוה במיוחד ויש להתחשב בכך. דגמי המדפסות לעבודה מאומצת מיועדים לפעולה רציפה במשך מספר רב של שעות בכל יום ומתאימים למשרדים וארגונים המדפיסים כמות חומר רבה.

התכונה	הקשה	מטריצה	לייזר
איכות הדפסת תווים	מצוינת	בינונית	טובה
מהירות הדפסה	איטית	בינונית	מהירה
תווים לשנייה (CPS)	עד 80CPS	עד 400CPS	4-10 דפים
שינוי אותיות ותכונות הדפסה	מעט	הרבה	הרבה
אפשרות לשינוי גופנים	החלפת מניפה - מסורבל	החלפת EPROM - טכנאי	טעינת תוכנה
יכולת הצגה גרפית	לא	כן	כן
רעש סביבתי	הרבה	בינוני	שקטה
הדפסת דפים בדידים	כן*	כן*	כן
הדפסת נייר רציף	כן	כן	לא כן*

* דרוש התקן מיוחד

השוואת תכונות של סוגי מדפסות

4.3.2 פרמטרים של מדפסות

פרמטרים שונים אשר נגזרים מתכונות המדפסת קובעים את האופן שבו יופיע התו המודפס על גבי הנייר. יצרני המדפסות השונים קובעים אילו פרמטרים ניתן לשנות בשינויי חומרה בלבד, אילו פרמטרים ניתן לשנות על ידי פקודות מהמחשב ואילו פרמטרים ניתן לשנות גם בחומרה וגם בתוכנה. אנו נתייחס לשינוי פרמטרים בתוכנה, כי בשינוי הפרמטרים בחומרה אין סטנדרט אחיד. הדבר כפוף לתכנון של היצרן וזאת יש לבדוק בספר היצרן של כל מדפסת. גם הפרמטרים להדפסה שניתנים לשינוי באמצעות פקודות הנשלחות מתוכנת המחשב אינם זהים בין כל המדפסות, אך קיימת אחידות די גבוהה בין היצרנים השונים, או לפחות גישה דומה לענין זה.

כאשר רוצים לשנות פרמטר הדפסה כלשהו, חשוב לבדוק אם יצרן המדפסת איפשר שינוי זה בתוכנה. במשך הזמן נוצרו מספר תקנים של פרמטרים להדפסה, כשהנפוץ ביותר הינו **Epson Mode** ולכן מרבית המדפסות מספקות חיקוי בתוכנה לפעולת מדפסת Epson. כתוצאה, כל פקודות הבקרה המתאימות למדפסת Epson יפעלו על כל מדפסת המאפשרת חיקוי למדפסת זו. למדפסת אשר אינה תומכת בחיקוי Epson יש צורך לבנות קובץ פרמטרים Printer Driver המאפשר לכוון את פעולתה. תכנית אשר אינה "מכירה" מדפסת מסוימת לא תוכל לשלוט על פעולתה כראוי ולנצל את כל תכונותיה בצורה טובה.

שינוי פרמטרים של המדפסת מתבצע על ידי שידור של של קוד בקרה למדפסת. זו מזהה את הקוד ומבדילה בינו לבין התווים המיועדים להדפסה. חלק מקודי הבקרה הינם צירופים מטבלת ASCII וחלקם – רצף של מספר תווים הנקראים **רצף בקרה** (Escape Sequence). רצף הבקרה מתחיל לרוב בקוד הבינארי של התו Escape (27 DEC) שבטבלת ASCII. כאשר המדפסת מזהה התחלה של רצף בקרה, היא מתייחסת לתווים שאחריו כהוראות לביצוע. תנאי הכרחי הוא, שהפקודות תהיינה כלולות באוסף הפקודות אשר המדפסת מכירה. הפרמטרים השימושיים אשר ניתן לכוון:

* מירווח בין תווים בשורה (צפיפות הדפסה CPI – Characters Per Inch) הצפיפויות הנפוצות: 5, 6, 8, 10, 12, 17 CPI.

* מירווח הדפסה בין שורות (צפיפות אנכית LPI – Lines Per Inch) צפיפויות קיימות: 6, 8 LPI.

* איכות הדפסה - הדפסה רגילה, או הדפסת איכות (NLQ ,Draft).

* הדגשת תווים (Bold), קו מתחת לתווים (Underline), אותיות עליונות (SuperScript), אותיות תחתונות (SubScript), רוחב כפול (Double Width) ועוד.

* גרפיקה של נקודות (Dot Graphics).

* הטענת גופנים במדפסות לייזר (Fonts Down Load).

ותכונות רבות נוספות.

את צפיפות התווים (17, 10 תווים באינצ') וצפיפות השורות (6 או 8) ניתן לכוון גם באמצעות פקודות DOS:

```
MODE LPT1:Char,LPI
```

כאשר:

Char – מספר תווים בשורה. ערכים אפשריים: 80 (10CPI), או 132 תווים (17CPI).

LPI – מספר שורות באינץ'. ערכים אפשריים: 6 או 8.

דוגמה: הפקודה MODE LPT1:132,8
תעביר את המדפסת לצפיפות 17CPI ול-8 שורות באינץ'.

משמעות הקוד / פירוש	קוד התחלה	קוד סיום
הדגשה אופקית (Bold)	Esc E	Esc F
כתב מורחב 5 תווים לאינץ' (Elongated)	Esc W 1	Esc W 0
כתיבה בחצי שורה תחתון (Subscript)	Esc S 1	Esc T
כתיבה בחצי שורה עליון (Superscript)	Esc S 0	Esc T
הדגשה על ידי קו תחתי (Underline)	Esc - 1	Esc - 0
הדגשה אנכית (Vertical Bold)	Esc G	Esc H
אותיות נטויות (Italics)	Esc 4	Esc 5

* רוב המדפסות פועלות בקוד Epson, או מחקות אותו.

קודים נפוצים במדפסות Epson או תואמות

4.3.3 הדפסה גרפית במדפסת מטריצה

כאשר מעבירים את מדפסת המטריצה לאופן הדפסה גרפי, מאפשר הבקר למחשב לפנות לכל סיכה במטריצת הסיכות שבראש ההדפסה. הדפסה גרפית היא איטית מאוד, מכיון שכדי לבצע אותה, המחשב שולח בראש רצף הנתונים קוד בקרה כדי להעביר את המדפסת לאופן פעולה גרפי. מעתה, המדפסת מתייחסת לתווים המגיעים מהמחשב בצורה גרפית ומתרגמת כל סיבית לנקודה על גבי הניר. סיבית שערכה 1 מודפסת כנקודה שחורה וסיבית שערכה 0 אינה מודפסת. כל בית הנשלח למדפסת מוצג כקו אנכי קצר באורך 8 נקודות, אשר סיבית LSB שלו מייצגת את המיקום של הנקודה התחתונה בקו וסיבית MSB מייצגת את המיקום של הנקודה העליונה בקו.

לדוגמה: הבית שערכו FF-Hex יצייר קו אנכי שלם, כי יש בו 8 סיביות שערכן 1. בית שערכו 1Hex יגרום להדפסה של הנקודה התחתונה של אותו קו בלבד ובית שערכו 80Hex יציג את הנקודה העליונה של אותו קו.

ישנם צירופי בקרה גרפיים רבים המאפשרים שליטה על מספר רב של פרמטרים: צפיפות הדפסה, מהירות הדפסה, הדפסה של 8 או 9 סיכות בכל הדפסה וכו'. כדי לשלוט על פעולת המדפסת בצורה גרפית ללא ממשק מוכן מראש, צריך לבדוק בספר היצרן את אפשרויות הבקרה השונות. אנו ממליצים לא לפעול בדרך זאת. זהו בזבוז זמן, כי קיימים ממשקי הדפסה גרפיים, אשר מיועדים להקל על המשתמש בעבודתו.

כדי להדגים את עיקרון הפעולה הגרפי מצורפת דוגמה (PRGDEMO) בנספח תכניות הדוגמה. זוהי דוגמה לקוד בקרה המעביר את מדפסת Epson לאופן פעולה גרפי במהירות כפולה ובצפיפות כפולה (Esc Y nẃ....). הפרמטר nw הוא מספר ברוחב 16 סיביות המציין את מספר הבתים הבאים ברצף, שאת ערכם יש להציג בצורה גרפית על גבי הנייר. מהירות התנועה של הראש הכותב מוכפלת על חשבון הקטנת צפיפות ההדפסה.

4.3.4 מתאם מדפסת מקבילית (Parallel Printer Adapter)

מתאם המדפסת המקבילית הוא כרטיס מתאם, המאפשר לחבר מדפסת מקבילית למחשב האישי לשם הפקת מידע בדפוס. למחשב ניתן לחבר עד שלושה מתאמים: LPT1, LPT2, LPT3. הכרטיס המתאם בדרך כלל אינו כרטיס נפרד, אלא משולב עם המתאם למסך על גבי אותו כרטיס. ניתן למצוא מתאמי מדפסת מקבילית על גבי כרטיסי תיאום יעודיים נפרדים, או על כרטיסים משולבים המכילים בנוסף למתאם המדפסת גם קשר להתקני קלט/פלט נוספים.

תקשורת מקבילית היא פרוטוקול המהווה אוסף כללים ששני ההתקנים צריכים לקיים על מנת ליצור ביניהם תקשורת. בהדפסה מקבילית הכרטיס המתאם והמדפסת מאפשרים העברה של נתונים מהכרטיס למדפסת במהירות גבוהה יחסית (כ-1000 תווים בשנייה). הנתונים העוברים למדפסת הם ברוחב של 8 סיביות, אשר מתאים בדיוק ל-256 צירופי התווים השונים הקיימים במחשב (קוד ASCII). הפרוטוקול המקבילי מאפשר העברה של כל 8 סיביות הנתונים בבת אחת מהמתאם למדפסת. כל סיביות הנתונים עוברות במקביל זו לזו בצורה אמינה וללא שגיאות.

התקשורת המקבילית מתאימה ואמינה לטווח קצר בלבד (מספר מטרים) ולכן נמצא תמיד שהמדפסת המקבילית צמודה למחשב. בין מתאם המדפסת לבין המדפסת מקשר פתיל המעביר את אותות הבקרה והנתונים. כרטיס מתאם המדפסת ממופה במרחב הקלט/פלט בכתובות 378-37F Hex, או בכתובות 278-27F Hex.

4.3.4.1 החומרה וכתובות כרטיסי התיאום למדפסות

בתהליך BOOT המחשב בודק את החומרה המחוברת אליו ומזהה היכן נמצא הכרטיס המתאם למדפסת המקבילית. אם הוא מצא כרטיס מתאם אחד או יותר, הוא כותב את כתובותיהם הפיסיות באיזור זיכרון ה-RAM, אשר נקרא איזור המידע של ROM BIOS, החל מהכתובת 0040:08Hex (ראה נספח).

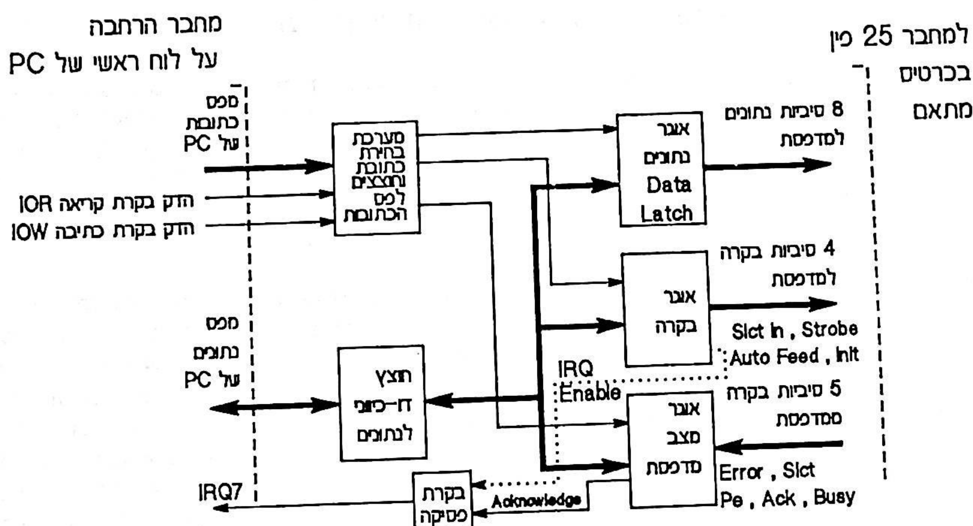
כתובת מתאם המדפסת היא ברוחב של מלה אחת (16 סיביות). כדי לשלוח נתונים ישירות למתאם, צריך לדעת את הכתובת ההתחלתית שלו.

כתובות המתאמים הן:

```
40:8 = LPT1
40:A = LPT2
40:C = LPT3
```

אם ערך הכתובת עבור מתאם המדפסת הוא 0, פירוש הדבר שלא מחובר כרטיס מתאם מדפסת למערכת המחשב. ערך הכתובת ההתחלתית של מתאם המדפסת דרוש על מנת לשלוח את הנתונים בצורה ישירה למדפסת (ראה תכנית לבקרת מדפסת בצורה ישירה בנספח תכניות הדוגמה). בכתיבה ישירה ניתן לשלוח נתונים מהתכנית למדפסת ללא שימוש בכלים של מערכת ההפעלה. בכתיבה ישירה משתמשים כאשר רוצים לחבר התקן חומרה כלשהו למתאם המדפסת, כמו לדוגמה תקע (plug) המשמש כמפתח הגנה בפני העתקות תוכנה.

הפעלה ישירה של המדפסת אינה מומלצת מכיון שהיא תלויה במעגלי החומרה של המחשב. על כן, תכנית הניגשת באופן ישיר למתאם המדפסת, לא בהכרח תפעל בצורה נכונה על מחשבים שיפותחו בעתיד! אפשרות נוספת הינה להפעיל את המדפסת המקבילית באמצעות פסיקות שירות של תכנית BIOS כפי שמוסבר בהמשך.



מתאם מדפסת מקבילית – תרשים מלבנים

4.3.4.2 האוגרים במתאם המדפסת

אוגר נתונים (Data Register) - אוגר ברוחב 8 סיביות המשמש לאחסון תו אחד שנועד להדפסה. אוגר הנתונים משמש כפלט מהמתאם למדפסת. כאשר תכנית רוצה לשלוח תו למדפסת היא צריכה להציב באוגר זה את ערך התו לפני שהיא שולחת למדפסת את ההוראות לקלוט אותו.

כתובת אוגר הנתונים היא 3BCHex או 378Hex, על פי הכתובת ההתחלתית של מתאם המדפסת. המספרים בתוך הריבועים שבשרטוט המצורף מציינים את מספר הפין במחבר המתאם. המספרים מעל הריבועים מציינים את מספר סיבית הנתונים באוגר הבקרה הקשורה לפין זה.

אוגר הנתונים במתאם המדפסת

	D7	D6	D5	D4	D3	D2	D1	D0
הדקי חיבור במחבר	9	8	7	6	5	4	3	2

כתובת אוגר הנתונים: כתובת בסיס של הכרטיס

אוגר בקרה (Control Register) - אוגר הבקרה שולח אותות בקרה למדפסת ומאפשר למחשב לשלוט על פעולתה. באמצעותו ניתן לבקש מהמדפסת לקלוט את התו הנמצא באוגר הנתונים, לבצע אתחול (Reset) וכו'. בשרטוט, המספרים שבתוך הריבועים מציינים את מספר הפין במחבר המתאם למדפסת. המספרים מעל הריבוע מציינים את מספר סיבית הנתונים באוגר הבקרה המחובר לפין.

אוגר הבקרה במתאם המדפסת

	D7	D6	D5	D4	D3	D2	D1	D0
הדק חיבור במחבר				EN	17	16	14	1

כתובת אוגר הבקרה: כתובת בסיס של הכרטיס + 2

אוגר מצב מדפסת (Printer Status Register) - המדפסת שולחת לאוגר המצב פירוט של מצבה. מתוך נתונים אלה יכול המחשב לבדוק אם המדפסת עסוקה, ללא נייר, סיימה לקלוט תו וכו' ויכול לקבוע אם אפשר לשלוח אליה נתונים להדפסה. בשרטוט המצורף, המספרים בתוך הריבועים מציינים את מספר הפין במחבר המתאם והמספרים שמעל הריבועים מציינים את מספר סיבית הנתונים באוגר המצב.

אוגר מצב מדפסת

	D7	D6	D5	D4	D3	D2	D1	D0
הדק חיבור במחבר	11	10	12	13	15			

כתובת האוגר: כתובת בסיס של הכרטיס + 1

4.3.4.3 הפינים במחבר המתאם וייעודם

אל מחבר המתאם מתחבר פתיל המדפסת, אשר כולל 25 פין D-Type מסוג נקבה. משמעות הדבר, שפתיל המדפסת חייב להיות מסוג זכר כדי שיוכל להתחבר אליו. פתיל המדפסת מתחבר למדפסת בצידו השני באמצעות מחבר בן 36 פין Centronoics מסוג זכר. הצורות השונות של המחברים במחשב מונעים בלבול בין מחבר המדפסת המקבילית לבין המחבר הטורי.

הטבלה הבאה מתייחסת לכרטיס המתאם. המונח "קלט" מכוון לקלט אותות המגיעים אל הכרטיס מהמדפסת (פלט של המדפסת) והמונח "פלט" הוא הנתונים הנכתבים מהמחשב לכרטיס המתאם וממנו למדפסת (קלט עבור המדפסת).

מספר פין	קלט / פלט	תיאור
1	פלט	אפשור מדפסת
2-9	פלט	סיביות נתונים D0-D7
10	קלט	אישור קליטת נתונים
11	קלט	מדפסת עסוקה
12	קלט	אין נייר במדפסת
13	קלט	מדפסת במצב שנבחר
14	פלט	קידום שורה אוטומטי
15	קלט	טעות במדפסת
16	פלט	אתחול מדפסת
17	פלט	בחירת כניסה
18-25	קלט / פלט	אדמה Ground

הפינים של פתיל המדפסת - צד מחשב

4.3.4.4 האותות במחבר של המתאם למדפסת

בפלט הבקרה מהכרטיס המתאם לכיוון המדפסת, הסימנים 0 ו-1 מציינים את ערך הנתון שנשלח על ידי המחשב, ולא את הנתון המגיע למדפסת. הכרטיס המתאם הופך את ערכם החשמלי של אותות הבקרה ולכן ההתייחסות היא לנתונים שצריכים להישלח על ידי המתכנת. לדוגמה, ערך 0 לבקרה שיישלח על ידי המתכנת יופיע כמתח חיובי עבור המדפסת!

אפשור מדפסת (Strobe) - במצב מנוחה הערך של **סיבית אפשור** (Enable bit) הוא 1. כאשר המחשב רוצה להורות למדפסת לקלוט נתון הנמצא על תילי הנתונים (ובתנאי שמדפסת פנויה) הוא יוריד את ערך סיבית האפשור ל-0 לפרק זמן מינימלי של מיקרו שנייה ויחזיר את ערך הסיבית ל-1. ירידה ועליה אלו של סיבית האפשור מורים למדפסת לקלוט את התו.

סיביות נתונים (Data 0 - Data 7) - סיביות אלו מכילות את ערך התו להדפסה. ערך התו חייב להימצא על תילי הנתונים לכיוון המדפסת לפני שמאפשרים קליטה בעזרת הדק אפשור מדפסת.

אישור קליטת נתונים (Acknowledge) - סיבית אישור קליטה נמצאת במצב 1 במנוחה. כאשר המדפסת סיימה תהליך קליטת תו, היא מורידה את ערך סיבית אישור קליטה למצב 0 למשך 5 מיקרו שניות ומחזירה את ערכה ל-1. תהליך זה מהווה אישור למחשב כי התו ששידר למדפסת נקלט על ידה. כמו כן משמש מנגנון זה עבור פסיקת מדפסת (Irq7 בבקר פסיקות, או פסיקה 0FHex במחשב). ברגע שמתקבל אישור קליטה המדפסת מתבצעת פסיקה למחשב, בתנאי שאפשרנו פסיקה, ואז הוא יכול לשלוח לה תו נוסף.

מדפסת עסוקה (Busy) - באמצעות סיבית זאת מודיעה המדפסת לכרטיס המתאם את מצבה. חיווי Busy אומר אם ניתן לשלוח למדפסת נתונים, או לא: כאשר המדפסת פנויה לקבל תווים הדק זה נמצא במצב 0, וכאשר המדפסת עסוקה ואינה יכולה לקבל תווים, היא תעלה את ההדק למצב 1. הדק זה הינו חיווי כללי למצב שבו המדפסת אינה יכולה לקלוט תווים והוא אינו מדווח את הסיבה. הדבר יכול להיות כאשר המדפסת עסוקה בקליטת תו, כאשר זיכרון המדפסת מלא (Printer Buffer Full) וזמנית אין היא יכולה לקבל נתונים נוספים, או כשיש בעיה אחרת שמונעת ממנה לקלוט תווים.

מדפסת במצב שנבחר - כאשר הדק זה במצב 0 לוגי, הדבר מסמן כי המדפסת נמצאת באופן שנבחר (Selected mode) והיא יכולה לקבל נתונים. כאשר הדק זה במצב 1, המדפסת אינה יכולה לקבל נתונים. כלומר, מצב 0 מסמן כי המדפסת מוכנה לעבודה.

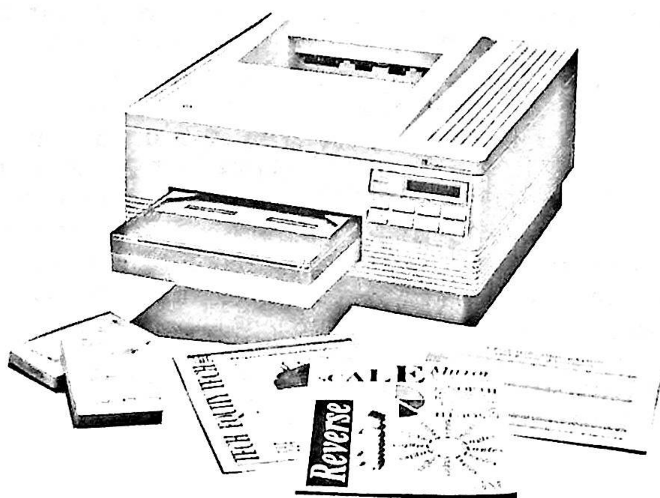
בחירת כניסה - על מנת שמדפסת תעבוד בתיאום עם המחשב, הדק זה חייב להיות במצב 0. כאשר הדק זה יועלה למצב 1, המדפסת לא תקלוט נתונים.

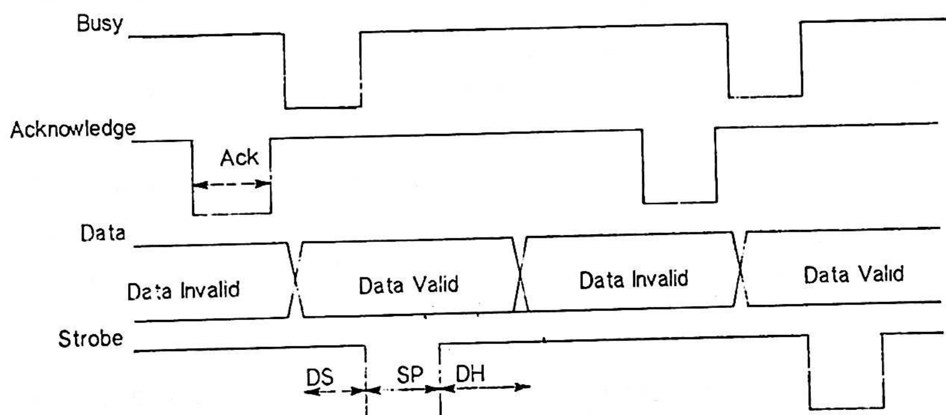
אין נייר במדפסת (PE) - במצב רגיל, כשיש נייר במדפסת, הדק זה במצב 0. כאשר הנייר במדפסת אוזל, הדק זה יעלה למצב 1 ויחד איתו תסמן המדפסת Error. הדק זה מאפשר חיווי מדויק של מצב המדפסת.

קידום שורה אוטומטי - במצב 0 בסיבית זו, המדפסת תקודם שורה באופן אוטומטי לאחר ההדפסה.

טעות מדפסת - ערכו של הדק זה במצב תקין של המדפסת הוא 0. ההדק יועלה על ידי המדפסת למצב 1, אם המשתמש צריך לגשת למדפסת לטיפול בתקלה. בכדי לבטל חיווי טעות מדפסת, על המשתמש לבצע פעולה מכנית/ידנית. לדוגמה: להזין נייר אם אזל, או לחבר את המדפסת אם היא מנותקת וכד'.

אתחול מדפסת - מצב רגיל הוא 1. כאשר נשנה מצב ל-0 נבצע אתחול למדפסת. הנתונים שהיו בזיכרון המדפסת (Printer Buffer) להדפסה יימחקו והיא תחזור ותכוון לכיוונים התחלתיים שהיו לה בזמן ההדלקה. כלומר, כל הפקודות לשינוי אופן העבודה שנשלחו אל המדפסת מתבטלות.





טבלת זמנים

DS – Data Set Up Time: Minimum 0.5 * 10-6 Sec
SP – Strobe Pulse Width : Minimum 0.5 * 10-6 Sec
DH – Data Hold Time : Minimum 0.5 * 10-6 Sec
Ack – Acknowledge Pulse Width : 5 * 10-6

תרשים זמנים - פרוטוקול תקשורת מקבילי

4.3.5 גישה בתוכנה למתאם המדפסת – פסיקות BIOS

0) שליחת תו למדפסת

בקריאה לפסיקת שירות, יש לציין את הפרמטרים הבאים באוגרים של המעבד על מנת לשלוח תו למדפסת המקבילית:

AH	=	0	שלח את הבית הנמצא באוגר AL אל המדפסת.
AL	=	בית	התו הנמצא באוגר זה יודפס.
DX	=	מספר	ערך בין 0 ל-2. מייצג את מספר המדפסת.
		מדפסת	LPT1, LPT2, LPT3.

בחזרה מפסיקה תעדכן תכנית BIOS את תוכן האוגרים הבאים:

AH	=	בית	בחזרה מפסיקה סיבית 0 של אוגר AH מחזירה 1, אם הבית לא נקלט במדפסת מסיבה כלשהי. שאר הסיביות מציינות את מצב המדפסת על פי הפירוט בסעיף "בדיקת מצב מדפסת".

1) אתחול מדפסת

בקריאה לפסיקה יש לציין את הפרמטרים הבאים באוגרי המעבד כדי לאתחל את המדפסת המקבילית (Reset):

AH = 1 אתחול מדפסת (Reset). כל הכיוונים שהתבצעו על ידי קודים שנשלחו מהתוכנה מתבטלים וזיכרון המדפסת מתרוקן מכל התווים הקיימים בו.

DX = מספר ערך בין 0 ל-2. מייצג את מספר המדפסת שיש לאתחל: מדפסת LPT1, LPT2, LPT3.

בחזרה מפסיקה תעדכן תכנית BIOS את תוכן האוגרים הבאים:

AH = מצב ערך אוגר AH מציין את מצב המדפסת על פי פירוט בסעיף "בדיקת מצב מדפסת".

2) בדיקת מצב מדפסת

בקריאה לפסיקה יש לכתוב את הפרמטרים הבאים באוגרי המעבד:

AH = 2 בדוק את מצב המדפסת והחזר את מצב הקו המקבילי.

DX = מספר ערך בין 0 ל-2. מייצג מצב של איזה מדפסת לבדוק: מדפסת LPT1, LPT2, LPT3.

בחזרה מפסיקה תעדכן תכנית BIOS את האוגרים:

AH = מצב ערך אוגר AH מציין מצב המדפסת.

מדפסת ערך 1 בסיבית האוגר מציין את מצב המדפסת כפי שמופיע באנגלית מימין לסיבית.

סיבית		תיאור
(Time Out)	0	מדפסת אינה מגיבה בתוך סביר. הדק Busy נמצא זמן רב במצב 0.
	1	לא בשימוש
	2	לא בשימוש
(I/O Error)	3	שגיאת קלט/פלט
(Selected)	4	מדפסת במצב שנבחר
(Out Of Paper)	5	אזל הנייר
(Acknowledge)	6	אישור קליטת בית על ידי מדפסת
(Not Busy)	7	המדפסת אינה פנויה

תכנית דוגמה לשימוש במדפסת ניתנת בנספח תכניות לדוגמה.

4.3.6 פקודת מערכת ההפעלה PRINT למדפסת המקבילית

הפקודה PRINT מאפשרת לבצע את הפעולות הבאות:

* קביעת פרמטרים של תכנית ההדפסה (קצב, מספר קבצים וכו').
* שליחת נתונים להדפסה.
* קבלת סטטוס של קבצים המחכים בתור להדפסה.
* מחיקת קבצים מתור ההדפסה.
* ביטול כל הקבצים בתור ההדפסה.

תבנית הפקודה:

```
PRINT [File-Name][/T][/C]
```

או

```
PRINT [/D:Device-Name][/B:Buffer-Size][/U:Wait-Time]
      [M:Print-Time][/S:Interval-Time][/Q:Queue-Size][/T][/C]
```

כאשר:

File-Name - שם קובץ/קבצים שיש לשלוח להדפסה. הקובץ/קבצים ייכנסו לתור הדפסה ויודפסו על פי סדר הכנסתם לתור.

/T - הוראה זו תמחק את שמות כל הקבצים המחכים בתור להדפסה ותרוקן בכך את התור למדפסת. קובץ הנמצא בתהליך הדפסה יופסק. בדרך כלל נראה כי המדפסת ממשיכה להדפיס, אך התוכן המודפס נמצא בזיכרון המדפסת ולא נשלח מהמחשב לאחר ביצוע הפקודה.

/C - הוראה לבטל את כל שמות הקבצים הבאים לפני פרמטר זה מתור ההדפסה. פרמטר זה חייב להופיע בלווית שמות קבצים לפניו.

את הפרמטרים הבאים ניתן להגדיר בהפעלה הראשונית של הפקודה PRINT. שינוי של פרמטר כלשהו מחייב תהליך BOOT חדש של המחשב.

Buffer-Size - גודל מאגר ההדפסה בזיכרון שיש להקצות לתכנית Print. ככל שמאגר ההדפסה גדול יותר, יהיו פחות גישות לדיסק במהלך ההדפסה. המחיר הוא בכך שנשאר פחות מקום בזיכרון עבור תכניות אחרות של המערכת. ערך ברירת מחדל המינימלית לזיכרון הדפסה הוא 512 בתים. ערך מירבי לזיכרון הדפסה הוא 1634 בתים. ניתן להגדיר פרמטר זה רק פעם אחת ויחידה בהפעלה הראשונית של פקודת PRINT. שינוי שרוצים לבצע מחייב BOOT למחשב.

Device-Name - מספר הערוץ המקבילי של המדפסת. ערכים אפשריים הם LPT3 ,LPT2 ,LPT1. פרמטר זה נוח כאשר מפעילים את פקודת PRINT בקובץ AUTOEXEC.BAT. הפרמטר פוטר מהצורך להקיש את שם הערוץ להדפסה בכל פעם שנבצע BOOT למערכת. ברירת המחדל היא ערוץ LPT1.

Interval-Time - פרמטר זה קובע את פרק הזמן שבו תתבצע פסיקה שתעביר פיקוד לתכנית PRINT. ערך ברירת המחדל הוא 8, כשערכים אפשריים לפרמטר זה הם בין 1 ל-255. יש לזכור, כי ככל שנאפשר מרווח זמן קצר יותר לתכנית PRINT נגזול יותר מזמן הביצוע של המחשב, אך כמות הנתונים שתודפס תגדל גם היא, בהנחה שהמדפסת מספיק מהירה על מנת לקלוט ולהדפיס את הנתונים.

Print-Time - ערך זה קובע כמה מחזורי שעון ניתן לחכות למדפסת בזמן הדפסה. ערך זה קובע כמה תווים רצופים ניתן לשלוח למדפסת כאשר מעבירים פיקוד לתכנית ההדפסה. ככל שמחכים זמן רב יותר, כמות התווים המודפסים תגדל ופרק הזמן המוקצב לביצוע תכניות אחרות יקטן. ערך ברירת המחדל עבור הפרמטר הוא 2, כשהערכים מותרים הם בתחום בין 1 ל-255.

Queue-Size - הפרמטר קובע את גודל תור הקבצים להדפסה. ברירת המחדל של המערכת היא שמירת תור של 10 קבצים. הערכים האפשריים הם בין 4 ל-32. הפרמטר מאפשר להגדיל/להקטין את מספר הקבצים המחכים להדפסה.

Wait-Time - הפרמטר קובע כמה מחזורי שעון יש לחכות לתגובה מהמדפסת עד אשר מחליטים שהיא אינה מגיבה, או אינה פנויה. יש משמעות לצפיה למדפסת, כי ייתכן שהיא תתפנה במשך ההמתנה. ערך ברירת המחדל הוא 1, כלומר בתשובה הראשונה שהמדפסת אינה פנויה חוזרים ולא ממתינים. ערך מירבי לפרמטר הוא 255. יש לזכור כי בזמן שתכנית PRINT ממתינה, תכנית המשתמש אינה מתבצעת, ולכן ההמתנה גוזלת זמן ביצוע מהמחשב.

הפרמטרים השונים קובעים את אופן פעולת תכנית PRINT עם מתאם המדפסת המקבילית. הערכים המוגדרים בהם קובעים את תצורת החומרה (שינוי שעון המערכת), תפוסת הזיכרון, כמות הזמן שתוקדש להדפסה ועוד. לא ניתן להגדיר רשימת ערכים הטובה לכל מקרה שהוא. כיוונון של ערכים אלה תלוי במהירות המדפסת, בסוג המחשב שאליו היא מחוברת ובאופי העבודות במחשב. כיוונון נכון עשוי לשפר את ביצועי מערכת המחשב והדפסת הקבצים. כדאי לנסות ערכים שונים ולבדוק את השפעתם על ביצועי המחשב.

תכנית PRINT היא תכנית תושבת, TSR, הנמצאת דרך קבע בזיכרון הראשי של המחשב (Resident) והיא מאפשרת למשתמש להדפיס נתונים תוך כדי ביצוע פעילות אחרת במחשב.

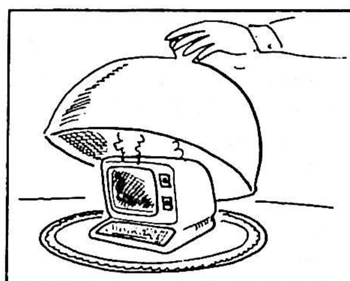
דוגמה:

הפקודה PRINT /D:LPT1/Q:20/B:1634

פקודה זו תטען את תכנית PRINT לזיכרון, תגדיר תור קבצים מירבי של 20 קבצים, מאגרים (Buffer) בגודל של 1634 בתים ותשלח את כל ההדפסות אל המדפסת המחוברת לערוץ המקבילי הראשון.

הפקודה PRINT HELP.LST

תשלח את הקובץ HELP.LST להדפסה במדפסת.



4.4 תקשורת אסינכרונית

4.4.1 למה תקשורת?

התקשורת האסינכרונית (Asynchronous transmission) הינה דרך נוספת לקשר בין המחשב האישי לבין המשתמש והמערכות הנמצאות בסביבתו. התקשורת מאפשרת להרחיב את אפשרויות הפעולה של המחשב האישי כדי להתחבר למחשבים אחרים ולהתקנים הפועלים בתקשורת. התקשורת האסינכרונית מאפשרת חיבור זול ופשוט, אם כי מוגבל במקצת. השימושים של תקשורת אסינכרונית הם:

* ממשק משתמש פשוט ונוח לתכניות יישומיות שונות. ממשק זה מתקבל באמצעות חיבור התקן "עכבר" לכרטיס התקשורת.
* חיבור כמסוף למחשב מרכזי ובכך לחסוך מסוף נוסף על שולחן המשתמש.
* חיבור בעזרת מודם למערכות מחשבים אחרות הפועלות בתקשורת זו.

כרטיס מתאם לתקשורת אסינכרונית חיוני כיום במחשב של כל משתמש המעוניין להתקשר בצורה נוחה אל המחשב האישי. ואומנם, מחשבים אישיים רבים מכילים כיום את הכרטיס הזה כחלק מהתצורה הבסיסית של המערכת. תכניות רבות יותר יודעות לנצל את צורת התקשורת האסינכרונית לצורך ממשק נוח לפעולה מול המשתמש.

תכניות הפועלות עם עכבר מאפשרות למשתמש לבצע את כל הפעולות בתכנית מבלי לגעת במקלדת. גם תכנית DOS Shell, המהווה חלק בסיסי של מערכת ההפעלה DOS גירסה 4 ומעלה, מאפשרת להתקשר באמצעות עכבר אל מערכת ההפעלה. תכניות יישומיות רבות אחרות משטח הגרפיקה, חלונות ועוד מאפשרות ממשק למשתמש באמצעות עכבר.

בפרק זה נעסוק בהיבטים של התקשורת הטורית האסינכרונית כממשק למשתמש. בפרק 7 נדון בנושא התקשורת אל הסביבה.

4.4.2 מתאם תקשורת אסינכרונית
(Asynchronous Communication Adapter)

הכרטיס המתאם לתקשורת אסינכרונית מאפשר למחשב האישי להתקשר להתקנים היקפיים המחייבים **תקשורת טורית** (Serial transmission). דוגמאות לכך הם עכבר, תוויין (Plotter), מדפסת רחוקה הפועלת בשיטה זו, מודם חיצוני ועוד. הכרטיס לתקשורת טורית בנוי סביב רכיב המסוגל לשדר ולקלוט נתונים בצורה אסינכרונית **בפרוטוקול Start/Stop** הקרוי UART (Universal Asynchronous Receiver/Transmitter). הפרוטוקול הינו הנוהל המוסכם, אשר כאן הוא מיושם לתקשורת.

הכרטיס המתאם לתקשורת משדר/קולט נתונים ברוחב 8 סיביות (בית) בצורה עצמאית. כל שנדרש מהמחשב הוא לכתוב את הנתון (בית אחד) אל הכרטיס, אשר יבצע את פרוטוקול ההתקשרות כדי להעביר את הנתון אל היעד. בקליטה, הכרטיס יקלוט את הנתון מהצד הרחוק ורק לאחר שסיים את קליטת הנתון, המחשב יידרש להתערב כדי לקחת אותו מהכרטיס. בגמר קליטה/שידור הכרטיס יבקש מהמחשב פסיקה לטיפול בתקשורת. המחשב יעבור לטפל בפניה של הכרטיס

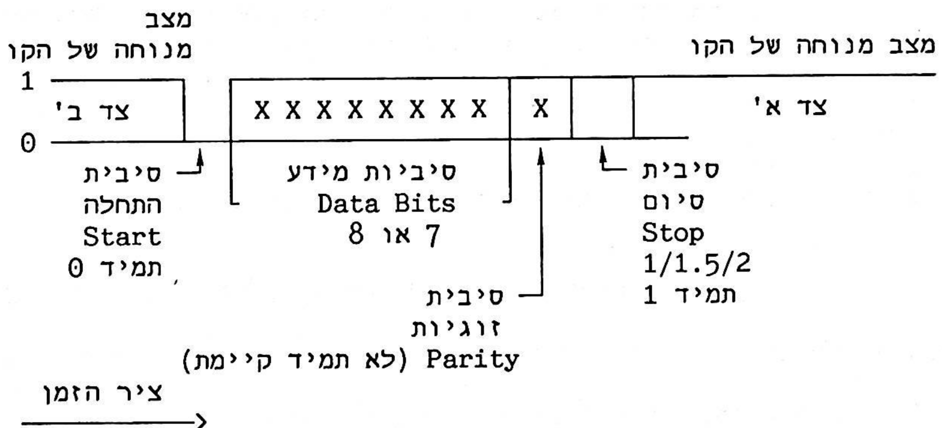
המתאם ובסיום הוא יחזור לטפל בתכנית שהוא מבצע, עד לבקשת הפסיקה הבאה מהכרטיס. דרך פעולה זו חוסכת מהמחשב את הצורך לטפל בכל שלבי ההתקשרות ומאפשרת לו לבצע אותה באמצעות מנגנון הפסיקה.

4.4.3 מהי תקשורת טורית?

תקשורת טורית (Serial transmission) הינה שיטת התקשרות שבה הנתונים עוברים בין שני התקנים בצורה טורית. כלומר, אם ברצוננו להעביר נתון ברוחב בית (byte) מהמחשב להתקן או להיפך, הסיביות יועברו בזו אחר זו (בטור) על גבי אותו קו תקשורת, כאשר בו-זמנית תופיע על קו התקשורת **סיבית אחת בלבד.**

פרוטוקול תקשורת טורי שונה מהפרוטוקול המקבילי, שבו כל שמונה הסיביות המרכיבות את הבית עוברות בו במקביל בו-זמנית. על קו התקשורת הטורי יש אות 1 לוגי, או אות 0 לוגי ברגע נתון (הנוהל קצת יותר מורכב בתקשורת דו-כיוונית). פרוטוקול טורי מאפשר להעביר את הנתונים למרחק גדול יותר מתקשורת מקבילית ונדרשים פחות תילים (חוטים) להעברתם.

פרוטוקול מקבילי, המקובל במדפסת, מחייב לחבר כבל של 25 תילים בין המדפסת לבין המתאם. פרוטוקול מינימלי לתקשורת טורית מחייב חיבור של 3 תילים בלבד! כדי ששני ההתקנים יוכלו להתקשר ביניהם בתקשורת טורית, עליהם להסכים מראש על מספר פרמטרים שיאפשרו זאת. בלעדי הסכמה, התקשרות בין ההתקנים אינה אפשרית.



תבנית הפרוטוקול הטורי האסינכרוני

4.4.3.1 תקשורת אסינכרונית

בתקשורת אסינכרונית **לא עובר אות סינכרון** רצוף בין שני ההתקנים, כי הסינכרון ביניהם נוצר מחדש בכל העברה של יחידת נתונים. יחידה זו מועברת במסגרת טורית (Frame), אשר מאפשרת סינכרון באמצעות סיבית Start וסיבית Stop (ראה תרשים). כל מסגרת משמשת להעברת יחידת נתונים אחת. הנתונים מועברים בקו מסגרת אחר מסגרת, אשר כל אחת מהן מהווה

התקשרות נפרדת. סיביות הסינכרון של הפרוטוקול הטורי האסינכרוני נתנו לו את השם "פרוטוקול Start/Stop". מקובל כיום בעולם המחשבים תקן אחד בתקשורת טורית אסינכרונית: **תקן RS232** בגירסה האמריקאית, או **תקן V24** - שמו הרשמי לפי תקן CCITT האירופאי.

צורת ההתקשרות הטורית האסינכרונית (ראה תרשים של תבנית הפרוטוקול):

במצב מנוחה, כאשר לא משודרים נתונים בין ההתקנים, הקו נמצא במצב מנוחה. מצב זה מוגדר על ידי העברת ערך 1 לוגי רצוף על קו התקשורת. כאשר צד א' מעוניין לשדר נתון, הוא מוריד את הקו למצב 0 למשך זמן של סיבית אחת. ירידת ערך קו התקשורת ל-0 למשך סיבית אחת מסמנת לצד ב', כי צד א' החל בהתקשרות. כלומר, זוהי **סיבית התחלה** (Start Bit).

ברגע שצד ב' זיהה תחילת התקשרות, הוא ממתין משך זמן של חצי סיבית. לאחר ההמתנה, בודק צד ב' את מצב קו התקשורת בזמנים קצובים באורך של סיבית אחת. על פי ערך הקו בבדיקה, מפענח צד ב' את תוכן הסיביות שנקלטו. דגימה/בדיקה של קו התקשורת במרכז הסיבית מאפשרת קליטה באמינות מירבית, גם אם יש הבדלי מהירויות קטנים בשידור או בקליטה בין הצדדים. בסיום שידור סיבית התחלה משדר צד א' את סיביות הבית זו אחר זו, החל בסיבית D0 וכלה בסיבית מידע אחרונה D7 או D6 בהתאם למה שהוסכם מראש (8 או 7 סיביות לבית בהתאמה). ערך סיביות המידע D0 עד D7 משתנה בהתאם לתוכן הנתון המועבר.

בסיום שידור סיביות הבית משדר צד א' **סיבית לבדיקת זוגיות** (Parity) שערכה נקבע ומחושב על פי ערכו של הנתון המועבר במסגרת. סיבית הזוגיות מאפשרת זיהוי שגיאות תקשורת והיא תחושב ותשודר רק אם הסכימו ביניהם הצדדים על העברתה (זהו חלק מהפרוטוקול). לאחר סיבית הזוגיות משודרת **סיבית סיום** (Stop). ערכה של סיבית הסיום תמיד 1 והיא מסמנת את סוף המסגרת. משך הזמן הדרוש לשידור של כל אחת מהסיביות במסגרת שווה וקבוע מראש, פרט למשך השידור של סיבית Stop שתלוי בצורת ההתקשרות. משך השידור (או "**אורך**") של סיבית הסיום יכול להיות כמשך השידור של סיבית מידע אחת, 1.5 סיביות, או 2 סיביות בהתאם לפרוטוקול ההתקשרות שנקבע.

4.4.3.2 פרוטוקול תקשורת טורית אסינכרונית

על מנת לקיים תקשורת טורית על שני הצדדים להסכים על **הפרמטרים** הבאים:

* **אורך סיבית** – אורך הסיבית קובע את קצב ההעברה (Bit Rate) של הסיביות מצד לצד, או כפי שמכנים זאת: BPS – Bits Per Second. קצב ההעברה הוא מספר הסיביות שניתן להעביר בין שני ההתקנים בשנייה אחת. במרבית האיזכורים בספרות המקצועית מכנים קצב זה בשם Baud Rate. מן הראוי לציין כי Baud Rate אינו קובע את מספר הסיביות שניתן לשדר בשנייה דרך קו התקשורת, אלא את מספר השינויים מ-0 ל-1 שניתן להעביר בין ההתקנים בשנייה.

בכל התקן קיימת אפשרות לכוון את רכיב UART לקצב ההעברה שהוסכם בין ההתקנים. כיוון קצב ההעברה נעשה בנפרד בכל צד, כך שכל התקן סומך על דיוק השידור של ההתקן האחר. מהירויות נפוצות לשידור נתונים

בתקשורת טורית הן 110, 300, 600, 1200, 2400, 4800, 9600, 19200 סיביות לשנייה.

* **מספר סיביות בבית** (Data Bits) - הפרמטר קובע כמה סיביות ישנן בכל מסגרת. קיימת אפשרות לבחור בין העברה של 7 סיביות לבין העברה של 8 סיביות בכל מסגרת. 7 סיביות מאפשרות להעביר 128 תווים שונים, שהם חלק ממגוון התווים האפשרי בשידור של 8 סיביות - 256 צירופים שונים אפשריים הכוללים גם גרפיקה ועברית.

* **זוגיות** (Parity) - סיבית הזוגיות מיועדת לגילוי שגיאות תקשורת בין שני הצדדים. מצבי פעולה אפשריים:

  * **ללא סיבית זוגיות** (No Parity) - לא מועברת סיבית זוגיות בין שני ההתקנים, אלא רק סיביות נתונים וסיביות התחלה/סיום. במצב זה לא ניתן לזהות ולגלות שגיאות בתקשורת.

  * **עם סיבית זוגיות** - עבור כל מסגרת (בית - byte) מועברת סיבית נוספת לבקרת שגיאות. בגמר העברת סיביות הנתונים במסגרת מחושב מספר הסיביות בבית ואז מוסיפים לבית סיבית אחת לביקורת זוגיות בהתאם לנוהל. אם נקבע שמספר הסיביות "1" צריך להיות **זוגי** (Even parity), תתוסף סיבית "1" רק אם מספר הסיביות "1" בבית היה לא זוגי. אם מספר הסיביות "1" היה זוגי, סיבית זו תהיה "0". אם נקבע שמספר הסיביות "1" צריך להיות **אי-זוגי** (Odd parity) תתוסף סיבית "1" כדי להשלים את מניין הסיביות "1" בבית למספר אי-זוגי. אם מספר הסיביות בבית הוא אי-זוגי, סיבית זו תהיה "0".

  דוגמה:
  כאשר סיביות הנתונים של הבית המועבר הן 01011101, יש בו 5 סיביות שערכן 1. אם נבחר בנוהל "זוגי", סיבית הזוגיות הנוספת צריכה להיות "1" כדי שמספר סיביות "1" במסגרת יהיה זוגי. כעת יהיו 6 סיביות של 1. אם נשלח את הצירוף 01100110, שכולל 4 סיביות "1" - תהיה סיבית הזוגיות "0", כדי לשמור על מספר זוגי של "1". אם בוחרים בנוהל "אי-זוגי", נוסיף סיבית "0" במקרה הראשון וסיבית "1" במקרה השני בהתאמה.

  הצד שקולט בית נתונים מחשב את סיבית הזוגיות בהתאם לנוהל שנקבע (זוגי/אי-זוגי) על פי הנתון שקלט ומשווה את ערכה המחושב לערך סיבית הזוגיות שנקלטה. אם ערך הסיבית שחושבה שווה לערך הסיבית שקיימת, סביר שהתקשורת נקלטה ללא שגיאות (סביר, אך לא בטוח!). אם ערך הסיביות שונה, יסמן כרטיס התקשורת הטורי שגיאה בקליטת התקשורת הטורית.

* **סיבית סיום** (Stop Bit) - סיבית הסיום מועברת בגמר שידור סיביות הנתונים של הבית וסיבית הזוגיות ומסמנת את סיום העברת המסגרת. אורך סיבית הסיום (כלומר משך השידור שלה בקו) יכול להיות מכפלה של 1, 1.5 או 2 של אורך סיבית הנתונים. סיבית הסיום משמשת להתקן הקולט זמן למנוחה ולהכנת המערכת לקליטת המסגרת הבאה. שים לב, מאחר ובמנוחה קיים ערך 1 על הקו, אורך סיבית הסיום קובע למעשה את הזמן המינימלי שבו יופיע 1 בסוף מסגרת. הזמן המירבי תלוי בזמן השידור של המסגרת הבאה. מכיון שאין מגבלה על משך הזמן המירבי בין סיום

שידור של מסגרת אחת לבין תחילת שידור של המסגרת הבאה, אורך סיבית "1" אינו מוגבל ומשתנה בהתאם לצרכים. דבר זה מאפיין את השידור האסינכרוני.

בפרוטוקול תקשורת אסינכרוני, המסגרת המשמשת להעברת הנתונים מכילה בנוסף לסיביות המידע, גם סיביות לבקרת מסגרת (התחלה, סיום, זוגיות). סיביות אלו גורמות לכך שקצב העברת הנתונים בקו נמוך יותר, כי עוברות בו גם סיביות מינהלה בנוסף לסיביות נתונים. מסיבה זו, המונח Bit Rate אינו חופף את המונח Data Bits Rate. חישוב מספר התווים המירבי אשר ניתן להעביר בשנייה דרך ערוץ תקשורת טורי, נתון על פי הנוסחה:

$$\text{תווים לשנייה BPS} = \frac{\text{Bit Rate}}{\text{Data bits} + \text{Stop bits} + \text{Parity bit} + \text{Start bit}}$$

דוגמה:
נתון פרוטוקול תקשורת אסינכרונית עם הפרמטרים הבאים:
(Baud) Bit rate = 4800
Data Bits = 8
Parity = Odd
Stop bits = 2

מספר התווים שיועבר בשנייה יהיה $\frac{4800}{8 + 2 + 1 + 1} = 400$

4.4.4 החומרה של מתאם התקשורת הטורית

כרטיס מתאם

כרטיס מתאם התקשורת מאפשר לשדר מידע טורי בצורה דו-סיטרית (Full Duplex), או בצורה חד-סיטרית (Half Duplex ,Simplex). התקשורת מתקיימת רק לאחר ששני הצדדים הסכימו על הפרמטרים לפרוטוקול התקשורת וכיוננו את כרטיסי התקשורת בהתאם. הנתונים שנשלחים לכרטיס התקשורת הטורי מהמחשב באמצעות תכנית ROM BIOS למשל, נכנסים לתוך אוגר שידור בתוך ה-UART שעל הכרטיס (ראה תכנית RSDDEMO בנספח).
ה-UART מבצע את הפעולה בשלושה שלבים:

שלב ראשון: מפנה את אוגר השידור לקליטת הנתון הבא ומעביר את הנתון הקודם לאוגר השידור הטורי (בהנחה שהוא פנוי).
שלב שני: משדר את הנתון בקצב ובמסגרת המתאימים (Start, Stop, Parity), בהנחה שהצד השני הודיע שהוא פנוי לקליטה.
שלב שלישי: מודיע למחשב שהינו פנוי.

הכרטיס הטורי מספק חיווי לתוכנה על מקום הימצאו של הנתון המשודר, באוגר שידור או באוגר שידור טורי. תהליך השידור כולל דו-שיח של אישורים בין הכרטיס המשדר לכרטיס הקולט, אשר מתקיימים באופן עצמאי על ידי הכרטיס הטורי ללא התערבות המחשב. תהליך השידור הטורי אפשרי רק

לאחר שקיים חיווי מהצד הקולט באמצעות הדק DSR, שהוא מוכן לקליטה. כאשר זיהה זאת, הכרטיס מבקש אישור לשידור מהצד הקולט באמצעות הדק RTS. לאחר קבלת האישור מהצד הקולט, באמצעות הדק CTS, מתחיל שידור המסגרת.

בתהליך הקליטה, המסגרת נקלטת באוגר קליטה טורי ורק לאחר השלמת קליטת המסגרת, היא (כלומר, הנתון) מועברת לאוגר קליטה ונמסר חיווי מתאים למחשב. הכרטיס הטורי גורם לפסיקה בגמר תהליך הקליטה, על מנת שהמחשב ייקח את הנתון הנקלט בזמן. אם הנתון הנקלט לא יילקח בזמן, ייתכן וייקלטו נתונים נוספים, אשר תוכנם "ידרוס" את הנתונים שקדמו להם והם יאבדו. אם אובדים נתונים, מסמן הכרטיס חיווי מתאים למחשב (OverRun). גם בתהליך הקליטה מעורב תהליך אישור ודו-שיח בין שני ההתקנים.

כרטיס התקשורת נותן חיווי באמצעות הדק DTR שהוא מוכן לקליטת נתונים, והוא מאשר באמצעות הדק CTS תחילת שידור מהצד האחר, ורק לאחר שנתבקש לעשות כן על ידי הדק RTS של הצד האחר. החיווי בתהליך הקליטה על מצב הנתון הנקלט מתייחס למצב הזוגיות, נכונות המסגרת ובתים שאבדו. את תהליך התקשורת הטורי אפשר לקיים בתוכנה בלבד, ללא דו-שיח בין ההתקנים ואישורים בחומרה לתחילת ההתקשרות. תהליך התקשורת בחומרה נקרא CTS, RTS ותהליך התקשורת בתוכנה נקרא XOFF, XON. בהתאמה.

כרטיס טורי אסינכרוני - תרשים מלבנים

עבור תקשורת בין התקנים, הכוללת אישורי חומרה דרושים מספר רב של תילים להעברת הנתונים, הבקרות והאישורים. בתקשורת בתוכנה, מסתפקים אך ורק בשלושה תילים. בתקשורת בתוכנה, צד אשר אינו מוכן לקבל מידע נוסף, שולח תו XOFF (צירוף 19 מטבלת ASCII). כאשר הצד השולח מזהה תו XOFF, הוא מפסיק לשלוח נתונים עד אשר הוא קולט תו XON מהצד המקבל (צירוף 17 מטבלת ASCII). פרוטוקול XOFF, XON הינו בעייתי, משום שאין הוא מספק חיוויים על מצב הקו. גם תהליך העברת הנתונים אינו שקוף וישנם צירופים שלא ניתן להעביר בקו, כמו למשל XOFF, XON המשמשים לבקרה. חוסר השקיפות של קו התקשורת יוצר בעיה לתוכנה, אך הוא נפוץ בשל החיסכון הגדול במספר התילים שיש להעביר בין ההתקנים.

4.4.4.1 קווי הנתונים והבקרה מהכרטיס

פרוטוקול תקשורת טורי אסינכרוני **RS232** מגדיר את כל הפרמטרים של התקשורת, אשר ניתן למצוא אותם גם בפרסומי CCITT תקן V24. הכרטיס המתאם התקני מתאים לרוב דרישות התקן. נדגיש כאן את הוראות התקן העיקריות:

- מצב 0 לוגי הוא מתח של 15+ וולט, הכרטיס מוציא רק 12+.
- מצב 1 לוגי הוא מתח של 15- וולט, הכרטיס מוציא רק 12-.

שים לב, שהמתח בפועל אינו מתאים לתקן, אך זהו למעשה מה שמקובל, ואין כאן כל תקלה בכרטיס.

התקשורת הטורית מתבצעת בין שתי תחנות **בלבד** והיא אינה מאפשרת לחבר מספר תחנות ברשת. אם רוצים לחבר מספר תחנות בחיבור טורי אסינכרוני, יש לחבר כל אחת מהן לכניסה נפרדת של **מאגד - RS232 MUX** (ראה הסבר להתקשרות באמצעות רשתות תקשורת ארציות ישראנת וסיפרנת).

על פי התקן, המרחק המירבי המותר בין שתי תחנות (האורך הפיסי של הקו) הוא 15 מטר. זהו המרחק המומלץ לתקשורת באמינות גבוהה, אבל במקרים רגילים ובתנאי סביבה שקטים מבחינה אלקטרו-מגנטית (lowRFI) שבה אין בסביבה מנועים גדולים או משדרי רדיו וכו', ניתן לחבר באמצעות פתיל תקשורת תקני מספר תחנות הנמצאות במרחק של עשרות או מאות מטרים ללא חשש לשגיאות.

הכרטיס מתחבר להתקן הטורי החיצוני של המחשב באמצעות מחבר זכר מסוג D-type הנמצא באחורי הכרטיס. קיימים שני סוגי מחברים: מחבר 25 פין על פי התקן, או מחבר 9 פין שאינו תקני ונמצא על גבי חלק מהכרטיסים.

בטבלה שלהלן נפרט את תפקיד ההדקים הפעילים של מחברי הכרטיס לתקשורת טורית. הסימון I מציין הדק "קלט" לכרטיס והסימון 0 מציין הדק "פלט".

מחבר 25 פין

מספר ההדק		השם	התפקיד
1		Ground	אדמה משותפת (0, אדמה מוגדרת של חברת החשמל)
2	0	TXD	**הדק שידור נתון טורי.**
3	I	RXD	**הדק קליטת נתון טורי.**
4	0	RTS	הדק בקשת אישור שידור (Request To Send).
5	I	CTS	הדק קבלת אישור שידור (Clear To Send).
6	I	DSR	מודם מוכן לפעולה (ההתקן הנגדי מוכן (Data Set Ready
7		SGnd	**אדמה משותפת לנתון.**
8	I	RLSD	מסמן שתדר נושא נקלט על ידי המודם. בקו תקשורת העובר דרך מודם, סימון זה מציין שיש מודם בקצה שני והוא פועל. (Received Line Signal Detect)
20	0	DTR	מציין לצד שני שכרטיס התקשורת פעיל ומוכן לפעולה (Data Terminal Ready).
22	I	RI	בפעולה עם מודם בקווי טלפון - מציין שיש צלצול מקו הטלפון ומאפשר למחשב להגיב לצלצול זה (Ring Indicator).

מחבר 9 פין

מספר ההדק		השם	התפקיד
1	I	DCD	מסמן שתדר נושא נקלט על ידי המודם. בקו תקשורת העובר דרך מודם, סימון זה מציין שיש מודם בקצה שני והוא פועל. (Data Carrier Detect)
2	0	TXD	**הדק שידור נתון טורי.**
3	I	RXD	**הדק קליטת נתון טורי.**
4	0	DTR	מציין לצד שני שכרטיס התקשורת פעיל ומוכן לפעולה (Data Terminal Ready).
5		SGnd	**אדמה משותפת לנתון.**
6	I	DSR	מודם מוכן לפעולה (התקן הנגדי מוכן (Data Set Ready
7	0	RTS	הדק בקשת אישור שידור (Request To Send)
8	I	CTS	הדק קבלת אישור שידור (Clear To Send)
9	I	RI	בפעולה עם מודם בקווי טלפון - מציין שיש צלצול מקו הטלפון ומאפשר למחשב להגיב לצלצול זה (Ring Indicator).

4.4.4.2 מבנה הכרטיס

הרכיב העיקרי בכרטיס הוא UART, זהו רכיב 8250 של Intel, או רכיב תואם לו. שאר הכרטיס מקשר את UART למחשב מצד אחד ולקו התקשורת מצד שני. כדי לחבר את הכרטיס לקו ישנם מתאמים מרמת מתח TTL (1=+5V, 0=0V) אל רמת המתח הנדרשת על פי תקן RS232 (1=-12V, 0=+12V). כדי לאפשר את פעולת הכרטיס הטורי בתוך המחשב האישי, קיימים על הכרטיס רכיבים

הממפים אותו בזיכרון הקלט/פלט (I/O) ומאפשרים לבצע פסיקת חומרה של ה-UART למחשב.

לרכיב UART יש שמונה אוגרים הממופים בכתובות קלט/פלט (I/O). באמצעות כתיבה וקריאה של תוכן האוגרים מנהל המחשב את הכרטיס הטורי. ניתן לחבר למחשב מספר כניסות טוריות אסינכרוניות, שכל אחת מהן נמצאת במחבר נפרד. לכל כניסה יש שם נפרד: כניסה טורית ראשונה נקראת COM1, כניסה טורית שנייה נקראת COM2 וכו':
כרטיס COM1 ממופה בכתובות קלט/פלט 3F8-3FF ויכול להפעיל פסיקת חומרה מספר 4. כרטיס COM2 ממופה בכתובת קלט/פלט 2F8-2FF ויכול להפעיל פסיקת חומרה מספר 3.

יש לזכור שכתובת 0 של הרכיב מתייחסת לכתובת הנמוכה של הכניסה האסינכרונית. לדוגמה, להפעלת כרטיס טורי COM1 בצורה ישירה, כתובת אוגר 0 של הרכיב היא כתובת 3F8Hex במרחב קלט/פלט וכתובת אוגר 8 של הרכיב היא 3FFHex.

בהמשך נראה ביצוע של תקשורת טורית בעזרת פסיקות שירות של תכנית BIOS (ראה סעיף תוכנה). תכנית אשר ניגשת ישירות לכרטיס מובאת בנספח תכניות הדוגמה. תכנית הדוגמה אינה פועלת בכל סוגי המחשבים ואינה מתאימה ליישומים מיוחדים הדורשים תקשורת טורית במהירות גבוהה, שאין אפשרות לבצעה דרך תכנית BIOS, כמו תקשורת טורית במהירות 110KBaud כפי שמעבירה תכנית Norton, למשל.

המשמעות של כל סיבית באוגרים הפנימיים של רכיב 8250 מופיעה בספר Micro Communication Handbook של חברת Intel, או בספר Technical Reference של יבמ.

4.4.5 חיבור התקנים לכרטיס התקשורת הטורית

את חיבורי ההתקנים השונים לכרטיס התקשורת הטורית האסינכרונית אפשר לחלק לשלוש קבוצות עיקריות. שים לב שמספר הפין מתייחס למחבר 25 פין תקני, ולא למחבר 9 פין.

שידור	קליטה	משותף
TXD 2	RXD 3	GND 1
RTS 4	CTS 5	SGND 7
DSR 6	DTR 20	
DCD-RLSD 8	RI 22	

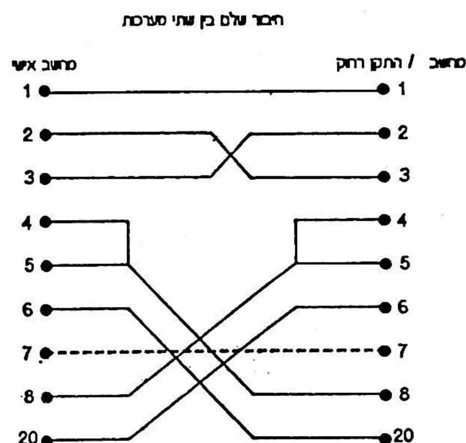
עד כאן הזכרנו שבתהליך התקשורת הטורית, ברגע נתון יש רק סיבית אחת בקו התקשורת. מצב זה אינו מדויק, מכיון שהתקשורת מסוגלת גם לקלוט וגם לשדר נתונים. כלומר, ברגע נתון יכולות להיות שתי סיביות בקו, אחת משודרת ואחת נקלטת. פעולות השידור והקליטה נפרדות לחלוטין מלבד פין 7 המשמש כאדמה המשותפת.

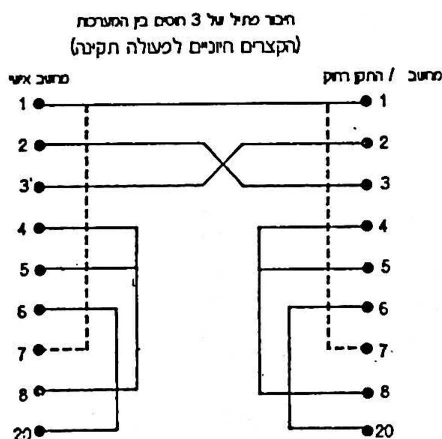
צורת חיבור של התקנים לכרטיס התקשורת הטורי נקבע על פי סוג ההתקן שיש לחבר ועל פי התכנית המפעילה את הכרטיס. בהתקנים המתחברים עם פתיל

עצמאי לכרטיס התקשורת הטורי, החיבורים בין הכרטיס להתקן כבר נעשו על פי הדרוש לתכנית המפעילה ולהתקן הפועל. ההסבר להלן מראה כיצד יש לחבר התקנים למחשב, בדרך שתאפשר למחשב להתקשר גם דרך תכניות מערכת ההפעלה (באמצעות פקודות PRINT, COPY) וגם דרך תכניות תקשורת למיניהן.

התקן המתחבר בתקשורת טורית למחשב, יכול עשות זאת באחת משתי דרכים אופייניות:

* **יציאה לתקשורת** - לדוגמה: יציאה למחשב אישי אחר או כמסוף למחשב מרכזי. פירוט הפינים השונים בהתקן זהה לפירוט הקיים בכרטיס הטורי. במקרה זה, יש לחבר את נקודות השידור והקליטה בין שני ההתקנים בהצלבה (שידור של הכרטיס לקליטה של ההתקן ולהיפך) ועל כן חיבור זה נקרא **חיבור מוצלב**.





תרשים חיבורים מוצלב

* **כניסה לתקשורת** - לדוגמה: כניסה ממודם המחובר לקו טלפון. כניסת TXD במודם מקבלת את יציאת השידור TXD של הכרטיס במחשב ומשדרת את הנתונים לקו הטלפון. בחיבור מודם יש לחבר במקביל את הנקודות בין המחברים במודם ובכרטיס המחשב. כלומר, יש לחבר את כל הנקודות אחת לאחת בין שני צידי הכבל שבין המודם למחשב.

לא ניתן לדעת מראש לגבי התקן מסוים אם החיבור הטורי לתקשורת הינו מסוג יציאה או כניסה. לדוגמה, מרבית המדפסות הטוריות מחוברות בכניסת תקשורת, כלומר הן מיועדות להתחבר למחשב ישירות בחיבור יציאת תקשורת. אבל יש מדפסות המיועדות להיות במרחק רב מהמחשב בחיבור דרך מודם. מכאן, כאשר נרצה לחבר מדפסת כזו ישירות למחשב (לא דרך מודם) נצטרך חיבור מוצלב.

4.4.5.1 חיבור מינימלי בין שני מחשבים

כפי שציינו קודם, אפשר לבצע חיבור טורי באמצעות שלושה תילים בלבד. יש לחבר את פין 7 של הכרטיס לפין 7 של ההתקן ולבצע הצלבה בין פין 2 לבין פין 3. זוהי הצלבה כפולה בין שידור לבין קליטה. במחבר של כל צד, יש לקצר בין רגליים 4 ו-5 ובין רגליים 6, 8 ו-20. פין 7 מחבר את האדמה המשותפת לשידור ולקליטה, אך לפעמים התקשורת תתקיים גם ללא חיבור. ללא חיבור זה ה"אדמה" שתהיה בשימוש תהיה "אדמה" של חברת חשמל, אבל הפגיעה באמינות התקשורת תהיה חמורה. חיבורי פינים 2 ו-3 הם חיבורי שידור, או קליטה. הקצרים נועדו ל"רמות" את המערכת. כלומר, אין קבלה אמיתית של אותות הבקרה מהצד השני והמחשב מקבל את אותות הבקרה שלו עצמו. בחיבור מינימלי יש סכנה של איבוד מידע, כאשר צד שני אינו מוכן לקליטה, או אפילו אינו פועל. כדי לפתור בעיית אובדן המידע נוצר פרוטוקול XON/XOFF, אך הוא גורם לבעיות אחרות כפי שציינו קודם.

המחשב האישי מאפשר להשתמש בכרטיס הטורי האסינכרוני למטרות רבות ומגוונות הקשורות בחיבורו למערכות אחרות. פירוט מלא של חיבורים אלה ניתן למצוא בפרק הדן ברשתות תקשורת.

4.4.5.2 העכבר — התקן המתחבר לכרטיס

העכבר, כהגדרתו בעולם החי, הוא חיה ממשפחת היונקים המכרסמים, בעלת זנב ארוך. בכל הקשור למחשב, העכבר עשוי מפלסטיק וזנבו הארוך הוא פתיל המקשר אותו למחשב האישי. העכבר משמש כממשק נוח לקלט בנוסף למקלדת. כאשר מניעים אותו על גבי משטח, הוא מעביר את תנועותיו ומיקומו למחשב על פי הסמן (בדרך כלל חץ) אשר מציין את המיקום על פני המסך. הזזה של העכבר על השולחן יוצרת תנועה מקבילה של הסמן על גבי המסך ומאפשרת לעקוב אחר תנועתו. לעכבר 3 לחצנים בדרך כלל, המשמשים לבחירה בין פונקציות שונות מתוך תפריטים המוצגים על המסך. כיום, יותר תכניות מכילות ממשק פעולה לעכבר וביניהן תכניות שירות שונות (תכניות מעטפת כמו DOS Shell של מערכת ההפעלה DOS גירסה 4 ותכניות גרפיות שונות).

תרגום תנועת העכבר על גבי המשטח לתנועה על גבי המסך יכול להיעשות בשני פתרונות חומרה עיקריים:

* **עכבר חיכוך:** כדור הנמצא בתוך "בטן" העכבר מתגלגל על השולחן כאשר מניעים את העכבר מצד לצד. תנועת הכדור מניעה שתי גלגיליות קפיציות המיועדות לתנועה אורכית ורוחבית של העכבר (שתי התנועות מאוזנות). סיבובי הגלגליות נספרים ומועברים למחשב כפקודות תנועה של העכבר ימינה ושמאלה, למטה ולמעלה.

  מצביע כדורי או **עכבר סטטי** הוא התקן ממשק מכני עם כדור בחלקו העליון. סיבוב הכדור מניע את הסמן. מלבד הבדלים מכנים והבדלי עיצוב, זהו עכבר חיכוך כדורי השוכב על גבו.

* **עכבר אלקטרו-אופטי:** שתי נוריות (Led) ושני התקנים לקליטת אור (פוטו דיודות) מותקנים ב"בטן" העכבר. העכבר נע על משטח מיוחד של לוח מבריק (ראי) אשר מחולק על ידי קווים אורכיים ורוחביים. קווי האורך והרוחב כהים יותר מהמשטח ואינם מחזירים אור ועל כן התקני

קליטת האור (פוטו דיודה) מעבירים למחשב את מספר הקווים האורכיים והאנכיים שנחצו על ידי העכבר. ספירת קווי אורך/רוחב שנחצו במהלך התנועה מאפשרת למחשב לעקוב אחר מיקום העכבר על גבי המשטח.

את **המשטח** המבריק יש להניח על השולחן בצורה שחלקו הארוך יותר נמצא במצב אופקי. כל אחד מהתקני קליטת האור מיועד לספור קווים מסוג אחד ואינו יודע לספור קווים מהסוג האחר. כלומר, התקן היודע להבחין בקו אורכי לא יזהה קו רוחבי ולהיפך.

יתרונו של עכבר האור האלקטרו-אופטי הינו בעיקר בהיעדר החלקים המכניים. הוא אמין יותר מעכבר חיכוך ואורך חייו גדול יותר. עם זאת, עכבר זה פחות נוח למשתמש, כי דרוש לוח מבריק מיוחד על מנת להניעו עליו. חסידיו של עכבר זה הופכים חיסרון זה ליתרון בטענה שהצורך בלוח מיוחד יגרום לפינוי מקום על השולחן כדי לאפשר את התנועה החופשית של העכבר.

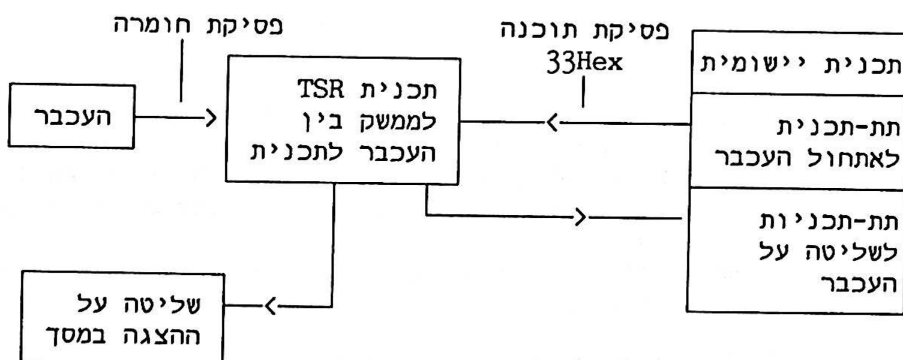
רגישותו של העכבר ודיוקו בתנועה נמדדים ביחידות "מיקי", כהוקרה למיקי מאוס. יחידת מיקי אחת מוגדרת כמספר ההודעות הנמסרות למחשב על תנועה במרווח 1/200 אינץ' של תנועת העכבר.

מרבית העכברים הקיימים בשוק מתחברים למחשב באמצעות הכרטיס הטורי האסינכרוני RS232, אך קיימים גם עכברים המתחברים לכרטיס מיוחד להם. בכל אחת מתצורות החיבור גורמת תנועת העכבר לפסיקת חומרה במחשב, אשר מאפשרת לעקוב אחר מיקומו. הנתונים שמעביר העכבר זמינים למתכנת באמצעות מנגנון פסיקת תוכנה 33Hex.

תכנית הממשק המקובלת ביותר בשוק לטיפול בעכברים הינה לפי התקן של מיקרוסופט (Microsoft), המוגדר בפירוט ב-MS Mouse Reference Guide. התקן מפרט שתי צורות גישה חליפיות לנתוני העכבר:

* **DOS Driver** - תכנית לטיפול בעכבר (קובץ MOUSE.SYS) הנטענת דרך קובץ CONFIG.SYS. תכנית זאת אינה נתמכת על ידי רוב היצרנים.

* **תכנית TSR** - גישה זאת נתמכת על ידי כל היצרנים. בגישת TSR נטענת תכנית TSR לזיכרון ומאפשרת ממשק בין התוכנה היישומית לעכבר.



סביבת הפעולה של העכבר במחשב האישי

לאחר טעינת תכנית הממשק לזיכרון המחשב, יכולה תכנית היישום להשתמש בעכבר על ידי קריאה לפסיקת שירות 33Hex. אוגר AX מכיל את סוג הפעולה שתכנית הטיפול בעכבר צריכה לבצע. כדי לפעול בתיאום עם העכבר, יכולה התכנית לברר את מיקומו, להחביא את הסמן שלו ולא להציגו על גבי המסך, להציג אותו שוב במיקום חדש, לברר את מצב הלחצנים השונים של העכבר, אם הם לחוצים או משוחררים. ראה תכנית בסיסית לניהול העכבר בנספח תכניות הדוגמה.

4.4.6 פקודות DOS לגישה לכרטיס תקשורת אסינכרונית

4.4.6.1 הפקודה MODE

למחשב האישי ניתן לחבר שני כרטיסי תקשורת אסינכרוניים אשר מוכרים למערכת DOS בשמות :COM1, :COM2. הפקודה MODE של מערכת ההפעלה DOS מאפשרת לכוון פרמטרים הקשורים בתקשורת טורית. מבנה הפקודה:

```
MODE COMn:BaudRate,Parity,Data-Bits,Stop-Bits,Retry-Action
```

כאשר:

n - מספר הכרטיס שיש ברצוננו לכוון (לכל כרטיס יש לתת הוראה נפרדת).
1 - COM1.
2 - COM2.

Data Bits - מספר סיביות הנתונים בתוך מסגרת (בית). ערכים אפשריים הם 7 או 8 סיביות.

Baud Rate - מהירות פעולה (Bit Rate). ערכים אפשריים:

מהירות	ערך בפקודה
110	11
150	15
300	30
600	60
1200	12
2400	24
4800	48
9600	96
19200	19

Parity - זוגיות. מכוון את סוג סיבית הבקרה לגילוי שגיאות. הערכים האפשריים הם:

N - No Parity
E - Even Parity
O - Odd Parity

Stop Bits - מספר סיביות סיום למסגרת. ערכים אפשריים: 1, 1.5, 2.

Retry Action - באיזו צורה לטפל בתשובה Busy של הכרטיס. כלומר, אם מנסים לשדר נתון ומקבלים מהכרטיס תשובה שלא ניתן לשדר, מה צריך לענות לתכנית הקוראת? ערכים אפשריים:

R - החזר הודעה שהכרטיס מוכן לשידור. המחשב יחזור ויפנה לכרטיס כדי לשדר נתונים, למרות שאין אפשרות לעשות זאת. פניה חוזרת לכרטיס צורכת משאבים רבים (זו ברירת המחדל של הכרטיס הטורי).

E - החזר קוד המסמן טעות (קוד שגיאה), אם הכרטיס מודיע שאין ביכולתו לשדר את הנתון.

B - החזר קוד המסמן שהכרטיס נמצא בתהליך שידור ואינו יכול לשדר נתונים נוספים (מתאים לערך P בגירסאות DOS שלפני גירסה 4).

לדוגמה: הפקודה MODE COM1:96,N,8,1,B תכוון את כרטיס התקשורת האסינכרוני הראשון (COM1) לפרמטרים הבאים: מהירות 9600 באוד, ללא סיבית זוגיות, 8 סיביות נתונים, 1 סיבית עצירה, החזר Busy אם כרטיס התקשורת עסוק ואינו יכול לשדר נתונים.

4.4.6.2 הפקודה COPY

פקודה COPY של מערכת ההפעלה מאפשרת לשדר קבצים בתקשורת טורית דרך הכרטיס האסינכרוני. כאשר משתמשים בפקודה זו המחשב עסוק כל הזמן בשליחת הנתונים ואינו יכול לבצע פעילות נוספת כלשהי. שליחת הנתונים מתאפשרת רק לאחר שמאתחלים את הפרמטרים להתקשרות בין הכרטיס לבין המדפסת הטורית, הן במחשב האישי והן במדפסת הטורית.

מבנה הפקודה: COPY File-Name COMn:

הפקודה תפעל באופן תקין רק לאחר שהפרמטרים לתקשורת בין הכרטיס להתקן המחובר אליו נקבעו בעזרת הפקודה MODE. הסבר הפרמטרים:

File-Name - שם הקובץ שיש לשדר דרך הכרטיס האסינכרוני.

n - מספר הכרטיס האסינכרוני (1 או 2).

הפקודה PRINT של מערכת ההפעלה מאפשרת להדפיס או לשדר קבצים באמצעות הכרטיס האסינכרוני. כאשר משתמשים בפקודה זו ניתן לבצע פעילות נוספת במחשב במקביל לביצוע ההדפסה. תהליך ההדפסה מורכב מהשלבים הבאים:

* הגדרת הפרמטרים להתקשרות לכרטיס האסינכרוני באמצעות הפקודה MODE.

* ביצוע הפקודה PRINT של מערכת ההפעלה. יש להפנות נתונים ל-COM1 או ל-COM2 במקום למדפסת מקבילית.

4.4.7 גישה לכרטיס באמצעות תכנית ROM BIOS

ניתן לבצע התקשרות בפרוטוקול תקשורת אסינכרוני באמצעות שפות תכנות שונות. בכל מקרה, הגישה לכרטיס הינה באמצעות פסיקת שירות מספר 14Hex של תכנית ROM BIOS המשמשת לתקשורת טורית RS232.

תיאור הגישה לכרטיס:

את הגישה לכרטיס מבצעים בעזרת מנגנון של פסיקת תוכנה בתכנית, אשר אפשרי בכל שפות התכנות. דוגמה לתכנית כזו ראה בתכנית RSBDEMO שבנספח תכניות הדוגמה. את הפרמטרים ואת בקשת השירות מתכנית ROM BIOS מעבירים באמצעות האוגרים שבמעבד. כאשר מנגנון הפסיקה מעביר את השליטה לתכנית ROM BIOS, התכנית בוחנת את ערכי האוגרים שהועברו אליה, מבצעת את בקשת השירות המתאימה ומחזירה את התשובה לתכנית הקוראת באמצעות אותם אוגרים של המעבד. כשהתכנית הקוראת מקבלת שוב שליטה, היא משתמשת בערכים החוזרים לצורך בקרה על פעולת הכרטיס והמשך התכנית.

0) כיוון פרמטרי התקשורת

בקריאה לפסיקה יש לשלוח את הפרמטרים הבאים על מנת לכוון את פרמטרי התקשורת השונים בפרוטוקול הטורי:

AH = 0 כיוון פרמטרים של פרוטוקול התקשורת בהתאם לתוכנו של אוגר AL.

DX = 0 פניה לכרטיס COM1.
= 1 פניה לכרטיס COM2.

AL = בהתאם לאופן הפעולה הרצוי

ערכי הסיביות באוגר

D1	D0	
		- מספר סיביות נתונים במסגרת.
1	0	- 7 סיביות (7 Data Bits).
1	1	- 8 סיביות (8 Data Bits).

D2	
	- מספר סיביות הסיום במסגרת (Stop Bits)
0	- 1 סיביות
1	- 2 סיביות

D4	D3	
		- סוג זוגיות (Parity).
0	0	- ללא סיבית זוגיות (No).
0	1	- זוגיות אי זוגית (Odd).
1	0	- ללא סיבית זוגיות (No).
1	1	- זוגיות זוגית (Even).

ערכי הסיביות באוגר

D5	D6	D7	מהירות (Baud Rate)
0	0	0	מהירות 110
1	0	0	150
0	1	0	300
1	1	0	600
0	0	1	1200
1	0	1	2400
0	1	1	4800
1	1	1	9600

בחזרה מפסיקה יעודכנו ערכי האוגרים הבאים:

AX = מצב   מצב הכרטיס האסינכרוני. ערך 1 בסיבית מציין שהתיאור מתאים למצב הכרטיס.

סיביות AH	תיאור
0	נקלט בית מצד רחוק.
1	נקלט יותר מבית אחד, אך מכיון שהמחשב לא קלט את הבתים שנקלטו בזמן, מספר בתים אבדו.
2	טעות זוגיות, הבית נקלט שגוי
3	טעות מסגרת, המסגרת שנקלטה איננה באורך המתאים (תיאום שגוי בין שני הצדדים).
4	נקלט צירוף של Break מצד שני.
5	אוגר שידור ריק, ניתן לשלוח לכרטיס נתון נוסף.
6	אוגר שידור טורי ריק (כל סיביות הנתון כבר נשלחו).
7	המידע לא נשלח בשל Time Out. תכנית BIOS מחזירה ערך זה אם הכרטיס איננו יכול לשדר.

סיביות AL	תיאור
0	
1	
2	
3	
4	התקבל אישור לשידור מצד שני (CTS).
5	צד שני מוכן לקליטה (DSR).
6	אבחנת טבעת (Ring Indicator).
7	נקלט אות מצד שני (במודם הדבר מציין שיש תקשורת בין הצדדים).

1) שליחת בית לצד שני

בקריאה לפסיקה יש לשלוח את הפרמטרים הבאים באוגרים על מנת לשלוח נתון דרך הכרטיס הטורי.

AH	=	1	שליחת הנתון הנמצא באוגר AL דרך כרטיס התקשורת הטורי לצד שני.
AL	=	הנתון לשידור	מכיל את ערך הנתון שיש לשדר.
DX	=	0	פניה לכרטיס COM1.
	=	1	פניה לכרטיס COM2.

בחזרה מפסיקה תעדכן תכנית BIOS את מצב אוגר AH.

AH	=	תיאור הפעולה	בחזרה מפעולה ערכו של אוגר AH מסמן הצלחת/כישלון הפעולה. אם ערכה של סיבית 7 באוגר AH הוא 1 הפעולה נכשלה. אם ערך הסיבית הוא 0 יש להתייחס לסיביות 6-0 כסיביות מצב קו תקשורת טורי, בהתאם לתיאור שהוסבר בפעולת כיוון פרמטרים.

2) קבלת נתון מכרטיס התקשורת

בקריאה לפסיקה יש לשלוח את הפרמטרים הבאים באוגרים על מנת לקלוט נתון מהכרטיס הטורי.

AH	2	קלוט נתון מכרטיס התקשורת לתוך אוגר AL.
DX	0	פניה לכרטיס COM1.
	1	פניה לכרטיס COM2.

בחזרה מפסיקת השירות מחזירה תכנית BIOS את הפרמטרים הבאים באוגרי המעבד:

AL	=	הבית הנקלט	תכנית BIOS מחזירה את ערך הבית שנקלט על ידי כרטיס התקשורת באוגר AL.
AH	=	תיאור הפעולה	אוגר AH מחזיר את מצב הפעולה ומצב שגיאות של קו התקשורת. ערכן של סיביות 1, 2, 3, 4, 7 באוגר הוא בהתאם להסבר שניתן בכיוון פרמטרים.

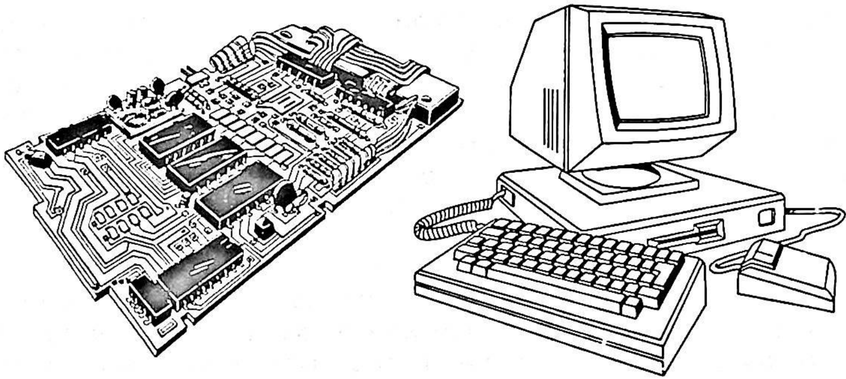
3) קבלת סטטוס של כרטיס תקשורת

בקריאה לפסיקה יש לשלוח את הפרמטרים הבאים באוגרי המעבד על מנת לברר את מצב הכרטיס הטורי:

AH	=	3	החזר מצב קו התקשורת.
DX	=	0	פניה לכרטיס COM1.
	=	1	פניה לכרטיס COM2.

בחזרה מפסיקת השירות תכנית BIOS מחזירה את הפרמטרים הבאים באוגרי המעבד:

AX	=	מצב הקו	מצב קו התקשורת. ערכי הסיביות באוגר בהתאם להסברים שניתנו בחזרה מפסיקת כיוון פרמטרים של קו התקשורת.



פרק 5

מערכת הדיסקים / דיסקטים במחשב האישי

מערכת הדיסקים/דיסקטים במחשב האישי קרויה גם **זיכרון משני** (Secondary storage). כשסקרנו את תכונות הזיכרון הראשי, **זיכרון RAM**, ראינו שיש לו מספר מגבלות: הוא אינו יכול לשמור נתונים לזמן ארוך מכיון שתכנו אובד בכיבוי המתח (מתנדף), גודלו מוגבל לצורכי עבודה תחת מערכת ההפעלה ומחירו גבוה יחסית. יתרונו הגדול של הזיכרון הראשי הוא במהירות ההעברה הגבוהה של הנתונים בינו לבין המעבד המרכזי.

הזיכרון המשני יכול לפתור את המגרעות של הזיכרון הראשי, על חשבון האטת מהירות הגישה. סדר גודל של זמן גישה במחשב של 10MHz לזיכרון RAM הינו $400x10^{-9}$ Sec (כלומר, פחות מחצי מיליונית של שניה) לעומת זמן של כ-30 אלפיות שניה (מילי-שניה) בדיסק קשיח, וזמן גדול פי 10 ויותר בכונן דיסקטים. כלומר, הזיכרון הראשי מהיר פי כמה עשרות אלפים מהזיכרון המשני.

הזיכרון המשני מורכב ממשטח בעל תכונות מגנטיות, אשר ניתן לכתוב עליו באמצעים חשמליים נקודות מיגנוט שהן סיביות הנתונים, וניתן לקרוא את הנתונים האלה על פי הצורך. את הנתונים ניתן לשמור לאורך זמן ולהשתמש בהם לפי צורך, כי ניתוק אספקת המתח אינו גורם למחיקתם. את הנתונים האגורים על גבי **המצע המגנטי** (Magnetic Media), או **דיסקה**, ניתן למחוק תוך כדי פעולה במחשב, באותם אמצעים חשמליים שמשתמשים בהם לכתיבתם. בפועל כותבים נתונים חדשים, או תווי "ריק" (תווי רווח או אפס למשל) על הנתונים הקיימים.

יכולת קריאה/כתיבה מקנה לזיכרון המשני גמישות רבה כאמצעי אחסנה של נתונים לזמן ארוך בקיבולת גבוהה ללא חשש מאובדן בעת ניתוק מתח החשמל, בצד מחיר אחסנה, אשר נמוך ממחירו של זיכרון RAM. קיבולת **הדיסקים הקשיחים** (Hard Disks) הינה עשרות, או מאות מגה בתים, כפי שאפשר לראות בדגמי הדיסקים השונים. לשם השוואה, הקיבולת של **תקליטונים**, או **דיסקטים** (Floppy Disks) הינה מ-360KByte עד 1.44MByte.

בהמשך, המונח "דיסק" או "דיסקט" יציין את סוג ההתקן המסוים בלבד. הכינוי "דיסק/דיסקט" יציין את שני הסוגים.

5.1 המצע המגנטי ואופן הגישה לנתונים

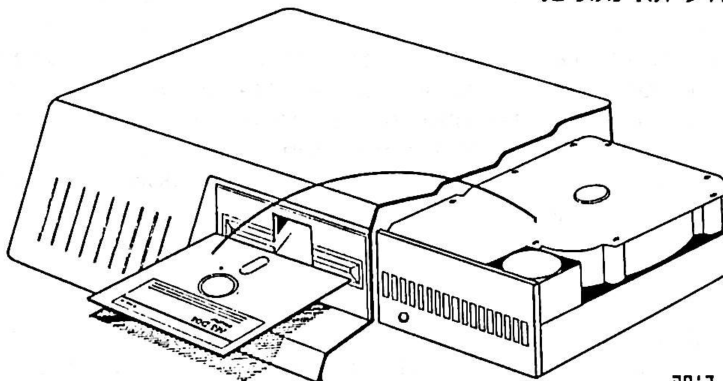
המצע המגנטי, בדיסק או בדיסקט, מורכב מדיסקית דו-צדדית המצופה בחומר בעל תכונות מגנטיות. על גבי הדיסקית ניתן לשמור נתונים ספרתיים בינאריים שמקודדים באמצעים חשמליים. את הנתונים ניתן לכתוב, לקרוא ולכתוב שוב באותם אמצעים. המצע שעליו רושמים את הנתונים הוא משטח מגנטי רציף מבחינה פיסית, אשר הכונן מחלק אותו **למסלולים** (Tracks) לוגיים בתהליך עריכה מיוחד. את הנתונים כותבים על גבי מסלולים אלה, שהינם בתבנית של מעגלים קונצנטריים בקטרים שונים.

מנוע צעד (Step motor) של הכונן מניע את הראש הקורא על פני המשטח ממסלול למסלול, עם רווח המפריד בין מסלול למסלול. המסלולים ממוספרים מ-0 ועד מספר המסלולים הכולל, פחות 1 (40 מסלולים: 0 עד 39).

5.1.1 כתיבת נתונים במצע המגנטי

בכל **צד** (Side) של הדיסקה המגנטית ישנו **ראש חשמלי** (Head) אלקטרו מגנטי המשמש לקריאה וכתיבה של נתונים. על פי כיוון הזרם החשמלי המוזרם דרכו בפעולת כתיבה, הוא גורם למיגנוט המשטח המגנטי שמתחתיו לצפון או דרום מגנטי. בקריאה, כאשר האלקטרו מגנט נע מעל המשטח המגנטי ומזהה שינוי מצפון לדרום או להיפך, הדבר גורם בו להיווצרות זרמים בקוטביות משתנה, אשר מתורגמים בחזרה לנתונים.

על פי חוק לנץ, הראש הקורא יכול להבחין רק **בשינוי** מגנטי. כלומר, ניתן להבחין רק במעבר מ-0 ל-1 לוגי, או להיפך. כדי לקבל את הנתונים בצורה נכונה, נבנים בתהליך האתחול של המשטח המגנטי שני מסלולים מגנטיים מקבילים זה לזה עבור כל מסלול נתונים יחיד. נהוג לומר, שמסלול מגנטי אחד הוא מסלול התזמון ומסלול מגנטי שני הוא מסלול הנתונים. מסלול התזמון בנוי מתאים, אשר נמצאים ברצף ותוכן כל אחד מהם הפוך לתא הקודם לו (ראשון 0, שני 1, שלישי 0 וכן הלאה). כאשר הראש הקורא חולף מעל מסלול התזמון הוא מזהה את השינויים המגנטיים בתאי התזמון ויכול להבחין בגודל תא נתונים.



כוננים במחשב

מכיון שהראש הקורא יכול לזהות רק שינוי בנתונים (צפון/דרום ולהיפך), הוא אינו יכול לזהות נתונים רצופים המכילים נתונים שווים. הוא סופר את מספר הפעמים של השינויים במסלול התזמון ועל פי מספר זה הוא קובע את התוכן הרשום במסלול הנתונים. מכאן שמסלול התזמון משמש כעין מונה קבוע לתנועת הראש הקורא ובלעדיו הוא לא יכול לזהות ולפענח את הנתונים שבמסלול הנתונים. מכיון שאין אפשרות לקרוא/לכתוב נתונים בצורה פשוטה למשטח מגנטי, קיים צורך להחליט על אלגוריתם ושיטה שבה הם ייקראו וייכתבו. שיטת הקידוד בין מסלול התזמון לבין מסלול הנתונים נקראת **אפנון נתונים** (Data modulation).

קיימות מספר שיטות לאפנון הנתונים ולממשק בין כרטיס בקר הדיסק לבין הדיסק המגנטי:

MFM - Modified Frequency Modulation.
RLL - Run Length Limited.
SCSI - Small Computer System Interface.
ESDI - Enhanced Small Device Interface.
IDE - Interface Design Enhancements.

שיטות האפנון נובעות מהטכנולוגיה שבה משתמשים לצורך ביצוע האפנון. שיטות אלו משפיעות על מהירות העברת הנתונים למשטח המגנוט וממנו, על בקרת שגיאות, אמינות וצפיפות הנתונים על גבי המשטח. השיטה הבסיסית לאפנון הנתונים היתה שיטת MFM. שיטת זו היתה בזבזנית במקום אחסון בגלל צפיפות נמוכה של הנתונים על פני המשטח המגנטי. יתרונה היה באפשרות ליישם אותה גם במשטח מגנטי בעל אמינות נמוכה יחסית.

במשך הזמן השתפרה הטכנולוגיה לייצור המשטחים המגנטיים ופותחו בקרים שפועלים בשיטת אפנון RLL. בשיטה זו ניתן לדחוס ב-50% יותר נתונים על גבי משטח מגנטי לעומת שיטת MFM. המשטח המגנטי הנחוץ לשיטת RLL חייב להיות באיכות גבוהה בהרבה ממשטח אפנון MFM, כדי שתהיה לו אמינות זהה ל-MFM. בשתי השיטות, הנתונים האנלוגיים עוברים באמצעות הכבל השטוח המוביל מכרטיס הבקר אל הדיסק. העברת הנתונים האנלוגיים בסביבה אלקטרונית עם "רעשים" סביבתיים גורמת להגבלת מהירות העברת הנתונים בין הדיסק לבקר שלו. המרת הנתונים ממבנה אנלוגי למבנה ספרתי ולהיפך מתבצעת בכרטיס הבקר.

התפתחות טכנולוגית נוספת הביאה לכך שניתן לבצע את ההמרה בדיוק גבוה באמצעות רכיב הנמצא על גבי הדיסק במרחק של עשרות מילימטרים בלבד מהראש הקורא ומהמשטח המגנטי. פיתוח טכנולוגי זה הביא לפיתוח שיטות SCSI ו-ESDI. שיטות אלו מיושמות בעיקר בדיסקים גדולים שהקיבולת שלהם מעל 70MByte. בדיסקים בקיבולת גבוהה דרושים משטחים מגנטיים בצפיפות גבוהה ומהירות העברת נתונים גבוהה ביותר. שיטה IDE שפותחה בזמן האחרון משלבת יתרונות טכנולוגיים ומחיר נמוך יחסית והופכת כיום לנפוצה ביותר בדיסקים קטנים.

מכאן והלאה לא נתייחס בנפרד למסלול התזמון ולמסלול הנתונים, אלא נראה את שניהם כמסלול לוגי יחיד המכיל נתונים. כמו כן לא נתייחס לשיטת אפנון המידע בתוך המסלול.

5.2 הדיסקט והדיסק המגנטי – מבנה ותכונות עיקריות

הדיסק הקשיח מורכב ממספר דיסקיות מגנטיות, אשר מורכבות אחת מעל חברתה עם מרווח מספיק לכניסת הראש ביניהן. כל הדיסקיות מסתובבות יחד והראש מוזז פנימה והחוצה באמצעות מנוע צעד.

כל המסלולים שיש להם קוטר שווה בכל משטחי המגנוט במערכת נקראים **צילינדר** (Cylinder). מסלול המידע הלוגי מחולק **לגזרות** (Sectors) לוגיות, אשר כל אחת מהן מקבלת מספר בין 1 עד למספר המקסימלי של הגזרות במסלול. כל גזרה מכילה 512 בתים של נתונים, אשר נרשמים בצורה סדרתית (בטור), סיבית אחר סיבית. כלומר, רוחב המסלול מבחינת נתונים הינו סיבית אחת, אך נהוג להתייחס ליחידת המידע הקטנה כסדרה של 8 סיביות המייצגות בית אחד. תנועת הראש הקורא על פני המשטח המגנטי מורכבת מצירוף של שתי תנועות:

* **תנועה אופקית** פנימה והחוצה (**חיפוש** – Seek) של הראש הקורא על גבי המשטח המגנטי. הראש הקורא "מרחף" על פני המשטח המגנטי ואינו נוגע בו! בכונן דיסקטים קיימים שני ראשים קוראים/כותבים ובדיסקים קיימים יותר ראשים קוראים/כותבים, כפול ממספר הדיסקיות המרכיבות את כונן הדיסק.

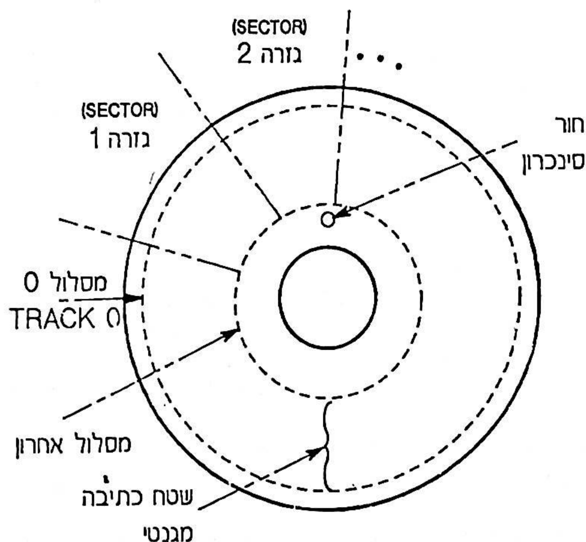
* **תנועה סיבובית** (Rotation) של הדיסק/דיסקט מביאה את הגיזרה הרצויה אל מתחת/מעל הראש הקורא. משך הזמן הדרוש להבאת הגיזרה מתחת/מעל הראש הקורא תלוי במהירות הסיבוב של הכונן. זוהי **ההשהיה של התנועה הסיבובית** (Rotational delay). מהירות הסיבוב של דיסקטים היא 300 סל"ד (סיבובים לדקה) ומהירות הסיבוב של הדיסקים היא בד"כ 3600 סל"ד, אך יש גם דיסקים המסתובבים במהירות גבוהה יותר.

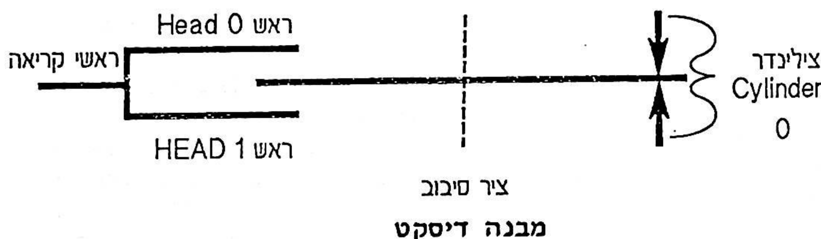
על מנת לקרוא נתונים האגורים על גבי המצע המגנטי, צריך הראש הקורא להגיע למסלול המתאים שעליו רשומים הנתונים בתנועה אופקית – Seek. תנועה זו מתאפשרת באמצעות מערכת של **מנוע צעד** (Step Motor) המניעה את הראש הקורא/כותב. המנוע מקבל פקודות תנועה מבקר הכוננים המחובר למחשב ומניע את הראש עד אשר הוא מיוצב מעל למסלול הנתונים המבוקש. זמן הגישה של הראש הקורא למסלול המתאים נקרא **"זמן חיפוש"** (Seek time). מהירות התנועה של מנוע הצעד אינה לינארית.

הזמן להנעת הראש הקורא ממצב מנוחה למסלול הראשון נקרא Track To Track Time. זמן תנועה ראשוני זה ארוך יותר מהזמן הנדרש להמשיך ולהניע את הראש למסלולים הבאים. ולכן, בין הפרמטרים המאפיינים את הכונן נמצא את הפרמטר המציין את **זמן החיפוש ממסלול למסלול** (Track To Track Seek) ואת **זמן החיפוש הממוצע** (Average Seek Time). זהו הזמן הממוצע הנדרש על מנת לחלוף על 1/3 של המסלולים בדיסקט/דיסק.

מנוע כונן הדיסקטים מסובב את הדיסקית המגנטית בתנועה סיבובית במהירות קבועה של 300 סיבובים לדקה. מנוע כונן הדיסק הקשיח מסובב את מערכת הדיסקיות של הדיסק הקשיח במהירות של 3600 סיבובים לדקה. סיבוב המשטח מאפשר לראש הקורא המתאים להגיע לגזרה המבוקשת במסלול, אשר חולפת מולו. לכל גזרה יש כתובת המאפשרת לזהות אותה ולכן, כאשר הראש הקורא

מתייצב במסלול המתאים והגזרה המבוקשת חולפת מולו במהלך הסיבוב, הוא מתחיל לקרוא את הנתונים ולהעביר אותם לבקר הכוננים ומשם בתהליך DMA לזיכרון המחשב. העברת הנתונים נעשית ישירות לזיכרון ללא התערבות המעבד).





מבנה דיסקט

כאשר מערכת ההפעלה רוצה לקרוא נתונים מהדיסקט, או מהדיסק, עליה למסור לבקר הכוננים **כתובת** (Address) של הגזרה שהיא מבקשת לקרוא. פעולה הפוכה מתבצעת בעת כתיבה. כתובת הגזרה מורכבת משלושה חלקים:

* מספר ראש קריאה (צד - Side).
* מספר מסלול (Track Number).
* מספר גזרה (Sector Number).

רק כתובת מלאה של הגזרה תאפשר למערכת ההפעלה לקרוא/לכתוב את הנתונים. מכיון שמהירות הגישה למשטח המגנטי איטית מאוד באופן יחסי למהירות הזיכרון הראשי במחשב, כל קריאה/כתיבה על גבי המשטח המגנטי תתייחס לגזרה שלמה (512 בתים) אחת לפחות. כלומר, גם אם נדרש לקרוא, או לכתוב בית בודד, יש לקרוא/לכתוב את כולה.

העברת הנתונים בין הבקר למחשב מהירה מאוד באופן יחסי לזמן הגישה (מאות קילו בתים בשנייה) ולכן יעיל יותר להעביר גזרה שלמה ולא בית בודד. העברה של הגזרה בשלמותה יעילה יותר, מכיון שבמרבית המקרים זקוקה התכנית, או מערכת ההפעלה גם לשארית הנתונים שבאותה גזרה. **קצב העברת הנתונים** (Data Transfer Rate) בין הבקר למחשב שונה במערכות

כוננים שונות ותלויה במהירות המחשב, סוג המעבד במחשב, סוג בקר הדיסק ושיטת אפנון המידע בדיסק.

דוגמה לפרמטרים אופייניים לדיסק קשיח:

זמן חיפוש ממסלול למסלול Track To Track Time - 20mSec מילי שנייה.
זמן חיפוש ממוצע Average Seek Time - 70mSec מילי שנייה.
מהירות סיבוב של הדיסק Speed - 3600RPM סל"ד.

מכאן, שזמן הגישה המינימלי למסלול בדיסק שבדוגמה הינו 20mSec וזמן הגישה המירבי כדי לחלוף על פני כל המסלולים (Full Stroke) הינו 170mSec:

```
Full Stroke = (Average-Time - Track-To-Track)x3 - Track-To-Track
   170mSec = (      70     -       20       )x20         3
```

5.3 דיסקטים ודיסקים — כוננים ובקרים

5.3.1 כונן הדיסקטים ובקר הכוננים

כונן הדיסקטים מאפשר לחבר למחשב האישי דיסקטים ולכתוב/לקרוא מהם נתונים. הכונן הוא מערכת מכנית שעליה מורכב כרטיס אלקטרוני, שתפקידו לתאם בין הכונן לבין כרטיס **בקר כונני הדיסקטים** – **FDC** (Floppy Disk Controller). השימוש בדיסקטים נוח למדי. את הדיסקט ניתן להחליף במהלך העבודה במחשב, לשאת אותו לכל מקום כדי להשתמש בו במחשב אחר, או כדי להעביר נתונים בין מחשבים. באופן זה ניתן להזין למערכת המחשב תכניות שונות שנרכשו על ידי המשתמש. נתונים מגנטיים אגורים על גבי הדיסקטים בצורה שהוסברה בפרק קודם, לפי צדדים, מסלולים וגזרות. בשוק המחשבים קיימים כיום שני סוגי כונני דיסקטים עיקריים: **5.25** אינץ' ו-**3.5** אינץ'. השמות לכוננים ניתנו לפי קוטר הדיסקית המגנטית שלהם. הדיסקטים נבדלים ביניהם, בנוסף לגודלם, גם במספר פרמטרים נוספים המפורטים בטבלה להלן:

קיבולת הדיסקט (Capacity) נתונה על פי הנוסחה:

מספר צדדים x מספר מסלולים x מספר גזרות x 512 (מספר בתים בגזרה)

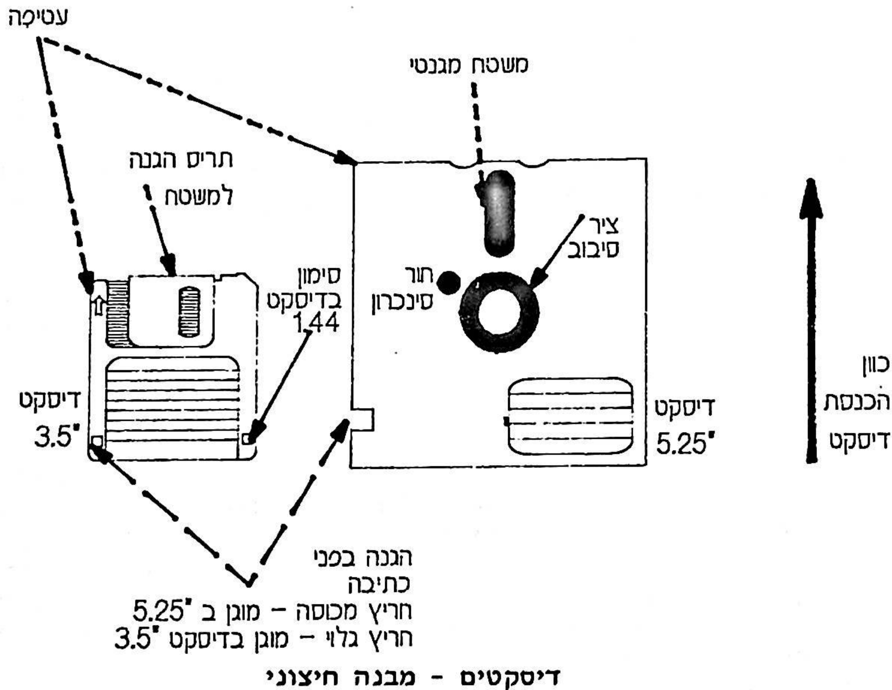
קוטר	קיבולת	מסלולים	גזרות
5.25	360KByte	40	9
5.25	1.2MByte	80	15
3.5	720KByte	80	9
3.5	1.44MByte	80	18

קיבולת ומבנה של דיסקטים שונים

5.3.1.1 חיוויים של כונן הדיסקטים

כונן הדיסקטים (Diskette Drive) מספק לכרטיס **בקר הכוננים** (Floppy Disk Controller) את **מצבו** (Status). הוא מסוגל לחוש את הדברים הבאים הנובעים מהפעלת כונן הדיסקטים:

* **מסלול 0** - לכונן ישנה מערכת חיווי המסמנת שהראש הקורא/כותב נמצא מעל מסלול 0. אות זה מסמן לבקר הכוננים כי הראש נמצא במסלול 0.



דיסקטים - מבנה חיצוני

* **הגנה בפני כתיבה** - קיימת אפשרות פיסית להגן על הדיסקט מפני כתיבה. החסימה נעשית באמצעות תווית הגנה בדיסקט "5.25 ובאמצעות פתיחת התריס הפלסטי בדיסקט "3.5. חיווי החסימה מנטרל את מערכת הכתיבה של הכונן. הוא מסמן לבקר הכוננים כי לא ניתן לכתוב נתונים על גבי הדיסקט. ההגנה הפיסית מאפשרת למשתמש למנוע מחיקת הדיסקט על ידי פעולה לא מבוקרת וטעות אנוש, או פגיעה בדיסקט על ידי וירוס כלשהו. לא ניתן לעקוף הגנה פיסית באמצעות תוכנה.

* **מהירות סיבוב** - הכונן מספק לבקר אות סינכרון בגמר השלמת סיבוב שלם של הדיסקט. באמצעותו הבקר יכול לעקוב אחרי השלמת המעבר על המסלולים השונים ואחר מהירות הסיבוב של הכונן. בכוננים שבהם מושג החיווי על ידי חור בדיסקט יש מערכת אופטית אשר מזהה אותו. הנתונים צריכים להיכתב על גבי המסלול במהירות סיבוב קבועה של הדיסקט – 300 סיבובים לדקה. קריאה/כתיבה לדיסקט במהירות שונה תגרום לשיבוש. כתיבה/קריאה במהירות שונה שקולה לשיר המוקלט על קלטת של רשמקול במהירות מסוימת ומושמע במהירות גבוהה, או נמוכה יותר. בשני המקרים

השיר יישמע מעוות ובמערכת מחשב הדבר קריטי יותר כי הנתונים לא יובנו.

* **חיווי פתיחת דלת** - רוב הכוננים מספקים חיווי אשר מאפשר לבקר להודיע כי דלת הכונן פתוחה ואין אפשרות לבצע פעולה כלשהי. ישנם מקרים שבהם יש חיווי על כך שבמהלך העבודה נפתחת הדלת או נסגרת.

* **חיווי סיבוב** - הכונן אינו מסתובב באופן קבוע כמו דיסק קשיח, אלא מתחיל להסתובב כאשר מתקבלת בקשת גישה. מכיון שכך, דרוש פרק זמן מסוים כדי להגיע למהירות הסיבוב הנכונה ופרק זמן נוסף כדי לעצור את סיבוב הדיסקט בגמר קריאה/כתיבה. חיווי על הסיבוב ועל הגישה לכונן ניתן למשתמש באמצעות הנורית הנדלקת בחזית הכונן. החיווי דרוש למשתמש כדי למנוע פתיחת דלת הכונן במצב של סיבוב, אשר עלולה לגרום לקריעת המשטח המגנטי או פגימתו.

בקר כונני הדיסקטים (FDC) הוא כרטיס מתאם אלקטרוני המתחבר על אחד ממחברי ההרחבה של הלוח הראשי ומאפשר לחבר למחשב האישי מערכות של כונני דיסקטים. לבקר כונני דיסקטים ניתן לחבר בדרך כלל שני כוננים, אשר יכולים להיות מסוגים שונים, "5.25 ו-"3.5. הכוננים מתחברים לבקר באמצעות כבל שטוח בעל שני מחברים המאפשרים חיבור זה. הכונן הראשון (A) מתקשר לכבל השטוח באמצעות המחבר המוצלב בקצה הקרוב למחבר. הכונן השני (B) מתקשר לכבל באמצעות המחבר המחובר אליו בצורה ישירה ללא הצלבות. ההצלבה בכבל שטוח המשמש לחיבור כונני דיסקטים היא בקצה הקרוב לנקודה מספר 1 בכבל, אשר ניתן לזהות אותה על פי צבעה האדום לעומת שאר חוטי הכבל השטוח שהם בצבע אפור.

במחשבים אישיים המבוססים על מעבד 8088/8086 הבקר לכונני הדיסקטים הוא כרטיס מתאם נפרד התומך בחיבור כונני דיסקטים "5.25 בקיבולת 360KB ובכונני "3.5 בקיבולת 720KB. התוכנה להפעלת הכוננים כלולה ב-ROM BIOS שעל גבי הלוח הראשי. אין אפשרות להגדיר את סוג הכונן המחובר למחשבים אלה, אלא רק את מספר הכוננים המחוברים למחשב. תוכנת ROM BIOS מכירה אך ורק כונני "5.25 בקיבולת 360KB. על כן, אם נחבר כונן של "3.5 בקיבולת 720KB מערכת ההפעלה תכיר אותו ככונן 360KB בלבד. כדי לגרום למערכת ההפעלה להכיר בכונן זה ככונן של 720KB ולהשתמש בו בצורה נכונה, יש צורך להגדיר אותו באמצעות קובץ CONFIG.SYS.

במחשבי AT ומעלה בקר הכוננים משמש גם כבקר למערכות הדיסקים הקשיחים. הוא מכיר ומסוגל להתחבר לכונני "5.25 ולכונני "3.5 מכל הסוגים. תכנית ROM BIOS מכירה את כל סוגי הכוננים ועל כן ניתן להגדיר תצורת מערכת מחשב באופן שמערכת ההפעלה תכיר את הכוננים השונים המחוברים לבקר ואת תכונותיהם.

את תצורת המחשב, אשר כוללת את תצורת הדיסקים/דיסקטים, שומרים על ידי כתיבה לרכיב CMOS עם גיבוי הסוללה שבתוך שעון זמן האמת. כיוונון מערכת הכוננים נקראת פעולת Setup. במחשבים חדישים מופיעה במהלך BOOT הודעת הדרכה מתאימה, המפנה לאפשרות SETUP. במחשבי AT ישנים יותר של יבם עושים זאת באמצעות תוכנת דיאגנוסטיקה מיוחדת היודעת לכתוב ל-RAM את תצורת המחשב הרצויה.

5.3.1.2 יתרונות וחסרונות של הדיסקטים

היתרונות:

* הדיסקטים נתיקים וקל להעביר באמצעותם נתונים ממחשב למחשב.
* פשוטים בהפעלה ותחזוקה, קל מאוד להכין אותם לפעולה.
* הדיסקטים זולים יחסית וקל להשיגם.

חסרונות:

* הדיסקטים נוחים להינזק (בייחוד דיסקטים גמישים של "5.25).
* קיבולת הדיסקטים קטנה יחסית (360KByte עד 1.44MByte). ישנם דיסקטים של 30MB אבל הם יקרים מאוד.
* הדיסקטים פועלים במהירות נמוכה. סדר גודל של גישה לנתונים – מאות אלפיות השנייה.

5.3.2 מערכת הדיסקים הקשיחים

5.3.2.1 הדיסק הקשיח ובקר הדיסק

כונן הדיסק הקשיח מורכב יותר מכונן הדיסקטים. את הדיסק הקשיח לא ניתן לשלוף מתוך הכונן, מכיון שהוא והכונן מהווים יחידה אחת בלתי נפרדת. הדיסק מורכב ממספר דיסקיות מגנטיות אשר לכל אחת מהן שני משטחי מגנוט, אשר לכל אחד מהם יש ראש קריאה/כתיבה נפרד. הדיסק הקשיח מסתובב כל הזמן במהירות גבוהה של 3600 סיבובים לדקה. יש כוננים שמהירות הסיבוב שלהם 5400 סל"ד.

זמן הגישה לנתונים בדיסק קשיח קצר יותר מזמן הגישה לנתונים בדיסקט. צפיפות המסלולים על גבי המשטח המגנטי גבוהה יותר וניתן לשמור יותר נתונים על גבי המשטח המגנטי. צפיפות המגנוט הגבוהה מתאפשרת מכיון שהדיסקית קשיחה ואטומה בתוך הכונן לשמירת יציבות וניקיון. הדיסק הקשיח מתחבר **לבקר הדיסקים** (HDC - Hard Disk Controller) באמצעות שני כבלים שטוחים בהתאם לשיטת האפנון. אל הבקרים הנפוצים של הדיסק הקשיח ניתן לחבר שתי יחידות דיסקים קשיחים.

במחשבים אישיים המבוססים על מעבד 8088/8086, התכנית להפעלת הדיסק הקשיח נמצאת על גבי כרטיס בקר הדיסקים ואין צורך להגדיר למחשב שדיסק כזה מותקן במערכת. בזמן BOOT, המחשב מזהה שקיים דיסק קשיח באמצעות רכיב ROM DISK הנמצא על כרטיס הבקר של הדיסק הקשיח. רכיב זה הינו תכנית, אשר מבצעת באופן אוטומטי אתחול של הדיסק ומכינה אותו לפעולה. במחשבי AT ומעלה התוכנה להפעלת הדיסק הקשיח כלולה ב-ROM BIOS ולכן יש להגדיר את סוג הדיסק בתצורת המערכת שב-CMOS RAM מגובה הסוללה.

קיימים סוגים רבים ושונים של דיסקים קשיחים, אשר מערכת ההפעלה ותכנית ROM BIOS צריכות להכיר את הפרמטרים השונים שלהם: מספר מסלולים, מספר גזרות במסלול, מספר ראשים קוראים ועוד. הגדרה לא נכונה של פרמטרים אלה תמנע אפשרות להשתמש בבדיסק.

הצילינדר האחרון בדיסק הקשיח אינו משמש לשמירת נתונים ואינו ניתן לשימוש מערכת ההפעלה. הוא משמש לבדיקות שונות של תקינות הדיסק. העובדה שאין עליו נתונים של המשתמש, מאפשרת לתכניות הבדיקה לכתוב

עליו ולקרוא ממנו ובכך לבדוק את תקינות הדיסק, מבלי לפגוע בנתונים של המשתמש. **אסור** שתכניות בדיקה כלשהן יבצעו בדיקת כתיבה על גבי הדיסק באיזור הנתונים! בדיקת כתיבה על מסלולי הנתונים תגרום למחיקתם.

תכנית FORMAT הכלולה במערכת ההפעלה DOS, אינה יודעת לבצע הכנה של הדיסק הקשיח לפעולה (הסברים מפורטים בהמשך הפרק). על מנת שהיא תכיר את המבנה הפיסי של הדיסק הקשיח יש לבצע תחילה תהליך הנקרא Pre-Format או **פירמוט חומרה.** לאחר מכן יש ליצור את **המחיצות** (Partitions) השונות בדיסק באמצעות תכנית FDISK של מערכת ההפעלה, ורק לאחר מכן לבצע FORMAT ב-DOS. תכניות דיאגנוסטיקה של יבמ מאפשרות Pre-Format כזה. קיימים גם כלי תוכנה נוספים למטרה זו כמו Disk Manager. תכנית Pre-Format לאתחול קיימת גם על גבי בקר הכוננים במחשב אישי מבוסס על מעבד 8088, או ב-ROM BIOS במספר מחשבי AT ומעלה. תכנית זו מוחקת את כל הנתונים הקודמים שהיו על גבי הדיסק הקשיח ורושמת את הגזרות מחדש.

כאשר מבצעים תהליך Pre-Format יש לשים לב לפרמטרים הנדרשים על ידי תכנית האתחול. פרמטרים שגויים עלולים לפגוע בביצועים של הדיסק.

5.3.2.2 פרמטרים עיקריים עבור Pre-Format

כאשר מבצעים Pre-Format יש להזין לתכנית את הפרמטרים הבאים:

* **מספר ראשים** קוראים/כותבים.

* **מספר צילינדרים** (Cylinders), או מספר המסלולים שעל פני המשטח המגנטי.

* **מספר גזרות במסלול** (Sectors Per Track).

* **פיצוי מראש** (PreCompensation) - מכיון שקוטר המסלולים הולך וגדל ככל שמתרחקים ממרכז הדיסק, צפיפות הנתונים במסלולים בעלי קוטר גדול יותר נמוכה יותר. במספר סוגי כוננים היצרנים ממליצים להכניס החל ממסלול מסוים, רחחו קבוע בין גזרה לגזרה, כדי להשיג צפיפות שווה של הנתונים על גבי המשטח המגנטי. באמצעות פיצוי מראש משיגים ביצועים טובים יותר של הדיסק.

* **רווח ביניים** (InterLeave) - מושג זה חשוב ביותר להשגת ביצועים טובים של הדיסק. רמת רווח הביניים קובעת כמה סיבובים יש לסובב את הדיסק מרגע שהראש הקורא נמצא מעל המסלול המבוקש עד שכל הנתונים הנמצאים על גבי המסלול יקראו על ידי המחשב. יוצא מכך, שאם המסלול יאותחל עם רמת רווח ביניים של 1, נשיג את קצב העברת הנתונים הגבוה ביותר (Data Transfer Rate). ברווח ביניים של 1, הגזרות מסודרות על גבי המסלול בזו אחר זו (3,2,1... ראה תרשים).

המחשב צריך לקלוט את הנתונים מבקר הדיסקים לזיכרון בקצב ההעברה של הבקר. אם הוא אינו יכול לעשות זאת, הראש יחלוף על פני הגזרה הבאה מבלי שיוכל להעביר לזיכרון את הנתונים שקרא, כי הבקר המתווך בינו לבין הזיכרון עסוק עדיין בהעברה של נתוני הגזרה הקודמת אל זיכרון

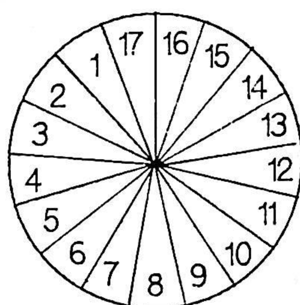
המחשב. במקרה זה נאלץ להמתין לסיבוב הבא של הדיסק על מנת לקרוא את הגזרה ש"חמקה".

אם נבצע Pre-Format עם רווח ביניים של 1 במחשב שאינו עומד בקצב הקריאה של הבקר, מהירות העברת הנתונים תהא נמוכה ולא תהיה התאמה בין הדיסק למחשב. לכן, הביצועים הטובים ביותר יושגו באמצעות התאמת רמת רווח הביניים בדיסק לקצב הקריאה של המחשב מהדיסק. ברמת רווח ביניים של 2, הגזרות אינן ממוספרות בזו אחר זו, אלא בדילוג ונוצר רווח של גזרה אחת בין גזרה לגזרה בעלת מספר עוקב.

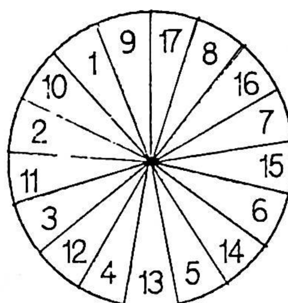
בתהליך Pre-Format נרשם בתחילת כל גזרה המספר הסידורי שלה במסלול וכך יודע הראש הקורא כי הגיע לגזרה המתאימה. בתהליך הקריאה קיים רווח בין גזרה לגזרה, ואם נבקש לקרוא גזרות 1 ו-2 מהמסלול, גזרת הביניים נותנת למחשב את פרק הזמן הדרוש לו כדי להעביר את הנתונים שנקראו אל הזיכרון. כך, הזמן החולף עד שהראש הקורא מתייצב מעל גזרה 2 מאפשר לנתונים של גזרה 1 להגיע לזיכרון המחשב והוא יכול להתחיל להעביר את הנתונים של גזרה 2. נעיין למשל בצורת הארגון ומספור הגזרות ברווח ביניים של 2 כאשר קיימות 17 גזרות במסלול:
משמאל (ראה תרשים): 1,10,2,11,3,12,4,13,5,14,6,15,7,16,8,17,9.

כלומר, אם נרצה לקרוא את כל הגזרות מהמסלול, נקרא בסיבוב ראשון את גזרות 1 עד 9 ובסיבוב שני של הכונן נקרא את הגזרות 10 עד 17. לכן רמת רווח זו נקראת רמת רווח 2. קיימות בשוק מספר תכניות המאפשרות לקבוע את רמת רווח הביניים האופטימלית לפעולה עם הבקר הנמצא במחשב ולבצע Pre-Format של הדיסק על פי רמה זו, מבלי לאבד את הנתונים הנמצאים בו (לדוגמה, תוכנת Spinrite או Norton5). **עם זאת, מומלץ לבצע גיבוי של הנתונים לפני שימוש בתכניות אלו ואחרות!** טכנאי מחשבים ומוכרים בחנויות לא תמיד מבצעים Pre-Format בצורה אופטימלית ועל כן מומלץ לבדוק אם רמת רווח הביניים מתאימה למחשב.

בין הכונן לבקר קיים שילוב מסוים הנוצר בתהליך Pre-Format. אם מחליפים את הבקר מסיבה כלשהי, יש צורך בדרך כלל לבצע שוב את תהליך Pre-Format. כי אם לא נעשה זאת, עלולה להיפגע מהירות העברת הנתונים



InterLeave = 1



InterLeave = 2

חלוקת הגזרות ברמות שונות של רווח ביניים

בין הדיסק למחשב וביצועי המחשב עלולים לרדת. כאשר מחליפים דיסק או בקר יש לוודא שרמת ביצועי הדיסק תואמת את יכולתו (שוב - ניתן לעשות זאת באמצעות כלי התוכנה שהוזכר קודם לכן). כמו כן, מומלץ לבצע גיבוי של הנתונים ולבצע Pre-Format אחת למספר חודשים ולאחזר את הנתונים שגיבינו. ביצוע התהליך יאפשר לשמור על תקינות הנתונים על גבי הדיסק וישפר את ביצועי המחשב. כאשר יוצרים קבצים חדשים על גבי דיסק שעבר הכנה מחדש לפעולה, הנתונים נרשמים בדיסק בצורה אופטימלית וביצועי המחשב יעלו (הסברים מפורטים לתופעה ניתן למצוא בהמשך).

קיבולת ב-MB	ראשים ק/כ	צילינדרים	גזרות במסלול	שיטת אפנון
10	4	305	17	MFM
21	4	615	17	MFM
32	6	615	17	MFM
42	5	977	17	MFM
42	6	820	17	MFM
80	9	1024	17	MFM
32	4	615	26	RLL
49	6	615	26	RLL
21	2	667	31	RLL
42	4	667	31	RLL

פרמטרים אופייניים לדיסקים קשיחים
(דוגמה בלבד)

5.3.2.3 יתרונות וחסרונות של דיסקים קשיחים

יתרונות:

* זמן גישה לנתונים בדיסק מהיר ביותר. זמן ממוצע 16 אלפיות שנייה בדיסק מהיר ועד כ-80 אלפיות שנייה בדיסק איטי.
* שימוש בדיסק קשיח חוסך את הצורך להחליף דיסקטים תוך כדי עבודה ומקל על עבודת המשתמש.
* קיבולת הדיסקים גבוהה ביותר, החל מעשרות ועד מאות מגה בתים. קיבולת גבוהה מאפשרת שמירת נתונים בצורה יעילה, נוחה וברורה על גבי הדיסק (ארגון בספריות, מחיצות וכו').

חסרונות:

* מחיר יקר יחסית לדיסקטים (מאות דולרים).
* דיסק קשיח אינו ניתן להעברה ממחשב למחשב, אלא לאחר פירוק.
* מחייב הבנה של המשתמש וידע על מנת לנצל את תכונותיו. דרושה הכנה לשימוש ואחזקה שוטפת.
* אובדן נתונים בדיסק קשיח פירושו אובדן כללי של הנתונים בכמויות גדולות אם אין גיבוי מתאים.

5.4 התייחסות מערכת ההפעלה DOS לדיסקטים ולדיסקים

מערכת ההפעלה צריכה להכיר את הסדר והמבנה שבו רשומים נתונים בדיסקט, כדי שתוכל לקרוא ולכתוב עליו נתונים. כדי להשיג את ה"היכרות הזו", יש להכין תחילה את הדיסקט לעבודה תחת מערכת ההפעלה DOS בפעולת **עריכה**, או **מיבנות** (Format), אשר נעשית באמצעות הפקודה החיצונית FORMAT. בתהליך זה שולחת מערכת ההפעלה לבקר הכוננים פקודות, אשר קובעות את החלוקה למסלולים (Tracks) וגזרות (Sectors). במהלך הסימון נערכת בדיקת תקינות של המשטח וסימון של האיזורים הלא תקינים. כלומר, מתבצעת כתיבה פיסית על גבי הדיסקט כדי לחלק אותו למסלולים וגזרות. כל גזרה מקבלת מספר המאפשר לזהות אותה ולכתוב/לקרוא ממנה. מספר הגזרה נע בין הערך 1 למספר הגזרות המירבי במסלול. בדיסק קשיח התהליך דומה וההבדלים יוסברו בהמשך.

5.4.1 כתובות והקצאת מקום בדיסק/דיסקט

לכל גזרה בדיסק, או בדיסקט, יש כתובת המאפשרת לזהות אותה. אבל בדומה למערכת הדואר, אפשר לקרוא לאותה גזרה במספר דרכים. נשווה את הכתובות בדיסק לכתובות דואר:

הגזרה היא הרחוב, המסלול - העיר, הצד - המחוז, שם הדיסק - הארץ. כמו בכתובת מגורים, יש גם מיקוד שהוא שיטה ספרתית עוקבת למציאה מידית של השכונה. השכונה המוגדרת במיקוד היא **Allocation Unit** או ***יחידת הקצאה***. שיטה זו מתייחסת לדיסק בסדר רציף עולה מבלי לציין צד, מסלול וכו' אלא רק את מספר ה-Allocation Unit. אפשר על כן, לציין מיקומה של גזרה מסוימת בשני אופנים:

* בהתייחס למיקומה הפיסי - צד, מסלול וכו'.
* בהתייחס למיקומה הלוגי - מספר יחידת הקצאה (Allocation Unit).

הערה: בספרות מצוין לעתים המונח **Cluster**, במקום **Allocation Unit**.

מערכת ההפעלה מחלקת את המשטח בצורה לוגית ל**יחידות הקצאה** – Allocation Units. כל יחידת הקצאה מכילה מספר שלם של גזרות, אשר גודל כל אחת מהן 512 בתים ומספרן במסילה שונה, בהתאם למבנה של הדיסקט/דיסק. כלומר, אין זה הכרחי שההקצאה של מקום בדיסקט תיעשה ביחידות פיסיות של גזרה אחת. מערכת ההפעלה יכולה להגדיר שכל קבוצה של 2, 4 או 8 גזרות תהיינה יחידת הקצאה לוגית אחת. יחידת הקצאה זו הינה יחידת השטח הקטנה ביותר המוקצית בתשובה לבקשת מקום על גבי הדיסק/דיסקט. לכן, כאשר מקצים מקום לשמירת נתונים בקובץ, ואפילו אם גודלו 10 בתים בלבד, יוקצה עבורו בדיסק/דיסקט שטח בגודל של יחידת הקצאה אחת שלמה. אם נדרש שטח גדול מיחידת הקצאה אחת – תוקצינה יחידות נוספות, תמיד בכפולה של מספר הגזרות שנקבעו.

הקביעה כמה גזרות (Sectors) תהיינה יחידת הקצאה (Allocation Unit) אחת משלבת יתרונות וחסרונות כתלות בגודל הקבצים השכיח במערכת. בקבצים בעלי קיבולת גדולה הכנסה של מספר גזרות רב ליחידת הקצאה אחת

(Allocation Unit גבוה) תתרום להקטנת פיזור הקובץ במקומות פיסים שונים על פני הדיסק ותבטיח זמן גישה קצר. באופן זה, הקובץ מאורגן ברצף גזרות והמרחק שהראש הקורא יצטרך לנוע על מנת לקרוא את תוכן הקובץ יקטן. מצד שני, בקבצים קטנים שגודלם פחות מיחידת הקצאה אחת, תגרום הקצאה של Allocation Unit גבוה לבזבוז רב, כי ההקצאה נעשית תמיד בכפולה של יחידות הקצאה. במקרה זה חלק מיחידת ההקצאה לא ינוצל, ואי אפשר יהיה להשתמש בו לקבצים אחרים במערכת.

יחידות ההקצאה בדיסקט ממוספרות החל ממספר 0. המספור מתחיל בראש 0 (כלומר, צד 0), מסלול 0, גזרה 1. כאשר יש יותר מגזרה אחת ביחידת הקצאה, יוקצו גם הגזרות העוקבות. המספור ממשיך במספרים עוקבים באותו צד ומסלול וממשיך לצד העוקב של אותו מסלול מגזרה 1 ובהמשך עובר למסלול הבא בצד 0 וכך הלאה, עד שמקצים את כל יחידות ההקצאה בדיסק/ דיסקט. ההקצאה נעשית על פי צילינדרים, כדי לגרום לתנועה מכנית מעטה ככל שאפשר של הראש הקורא ולהקטנת זמן הגישה לנתונים.

בתהליך Format על **דיסקט**, מערכת ההפעלה עוברת ורושמת את כל יחידות ההקצאה זו אחר זו ובכך הורסת את המידע שהיה על הדיסקט. בגמר תהליך Format, יוצרת מערכת ההפעלה את גזרת Boot, את ה-FAT ואת ספריית השורש (Root Directory). בגירסה 5 של DOS ניתן לבצע תהליך Format לא הרסני המאפשר גם שחזור של הנתונים בדיסקט.

בתהליך Format של **הדיסק הקשיח** מערכת ההפעלה אינה מחלקת את הדיסק למסלולים וגזרות ואין היא רושמת על גבי הדיסק נתונים כלשהם. הכנת הדיסק לפעולה וחלוקתו בצורה פיסית למסלולים וגזרות נעשית בתהליך Pre-Format, או Low Level Format (ראה הסברים בהמשך). תהליך Format אינו הרסני לנתונים שהיו קיימים בדיסק. מערכת ההפעלה רושמת רק את גזרת Boot, טבלאות FAT ואת ספריית השורש. בתהליך זה מכינה מערכת ההפעלה את שטחי העבודה שלה ובכך היא "מנקה" את הדיסק מנתונים קודמים. היא מנתקת קשרים קודמים שבין המדריכים לבין הנתונים, אך היא אינה מוחקת נתונים בפועל. כלומר, **ניתן לשחזר** את הנתונים אם ביצענו Format בשגגה על דיסק קשיח.

5.4.2 מבנה דיסק/דיסקט תחת מערכת ההפעלה DOS

גזרת Boot	Boot record
עותק ראשון של FAT	
עותק שני של FAT	
ספריית השורש	Root directory
איזור הנתונים - הקבצים	

כל דיסק, או דיסקט בנוי באותה מתכונת, ללא תלות בקיבולת שלו. גודל האיזורים השונים המוקצים לשימוש של מערכת ההפעלה משתנה בהתאם לסוג הדיסק/הדיסקט והקיבולת שלו.

5.4.2.1 גזרת BOOT

גזרת BOOT הינה תמיד על מסלול 0, צד 0 גזרה 1 של כל דיסקט שעבר תהליך Format. בדיסק קשיח נמצאת גזרת BOOT בגזרה הראשונה המוקצית לדיסק הלוגי, ולא בגזרה הראשונה של הדיסק (הסבר לכך תמצא בהמשך). בגזרה זו נמצא כל המידע הדרוש לתכנית BIOS בתהליך BOOT. בדיסקט הערוך לטעינת מערכת הפעלה, תכיל גזרת זו את כל המידע הדרוש לטעינת מערכת ההפעלה ובדיסקט שאינו ערוך לטעינת מערכת ההפעלה, מסומנת הגזרה בהתאם. אם ננסה לבצע טעינה דרך דיסק זה נקבל הודעת שגיאה: Non system disk.

גזרת BOOT הינה תמיד בגודל של גזרה אחת (512 בתים). המידע נרשם בטבלת פרמטרים BPB (Boot Parameters Block), אשר מפרטת את מבנה הדיסק/דיסקט כפי שרואה אותו מערכת ההפעלה. את ערכי טבלת BPB ניתן להציג בעזרת תכניות שונות (כמו Norton), אך לא מומלץ לשנות אותם. שינוי לא מבוקר עלול לפגוע בגזרת BOOT של הדיסקט/דיסק ומערכת ההפעלה לא תוכל להשתמש בהם לטעינה. למידע נוסף, ראה תכנית BOOT בנספח תכניות הדוגמה.

מערכת ההפעלה ניגשת למסלול 0 בכל דיסק/דיסקט בתדירות גבוהה מכיון שרשומים בו כל הנתונים החיוניים להפעלה, כמו FAT, ספריית השורש וגזרת BOOT. מסיבה זו סיכויו של איזור זה להיפגע גבוהים יותר מסיכוייו של איזור אחר בדיסקט.

5.4.2.2 פירוט הכתובות בגזרת BOOT

הכתובות בגזרת BOOT מפורטות החל מכתובת 0 (ערכים הקסה דצימליים).

כתובות (Hex)	פירוט הייעוד
0-2	3 בתים: כתובת קפיצה בתוך גזרת BOOT לצורך המשך טעינת מערכת ההפעלה DOS מדיסקט שניתן לבצע דרכו BOOT. בית ראשון EB-Hex מציין שפקודת הקפיצה בת 2 בתים: בית ראשון E9-Hex מציין כי פקודת הקפיצה היא בת 3 בתים (ראה ציור בהמשך).
3-A	8 בתים: גירסת מערכת ההפעלה ושם יצרן מערכת ההפעלה.

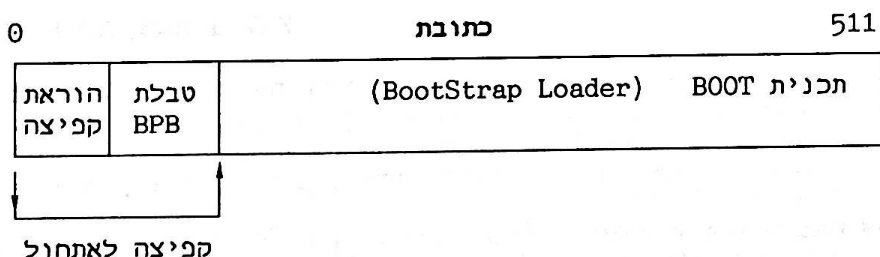
ערכים השייכים ל-BPB:

B-C	2 בתים (מלה אחת - Word): מספר הבתים בכל גזרה (Bytes/Sector).
D	בית אחד: מספר הגזרות ביחידת הקצאה אחת (Sectors/Allocation Unit).

E-F	2 בתים (Word): מספר הגזרות השמורות שלא ניתנות להקצאה לקבצים (בד"כ 1 עבור גזרת BOOT).
10	בית אחד: מספר טבלאות FAT. ישנם מקרים שבהם מערכת ההפעלה אינה משתמשת בעותק שני של FAT (כמו למשל, בדיסק המוחזק בזיכרון RAM).
11-12	2 בתים (Word). מספר הכניסות האפשריות בספריית השורש (ראה טבלת מספר קבצים מירבי בספריית שורש).
13-14	2 בתים (Word): את הערך 0 או את מספר הגזרות בדיסק לוגי (לדוגמה, בדיסק 360KB הערך יהיה 720, בדיסק עם מחיצות הערך יהיה כמספר הגזרות המוקצות לדיסק לוגי).
15	בית אחד המתאר את סוג המדיה המגנטית (ערכים אפשריים ראה בטבלת מספר קבצים מירבי בספריית שורש).
16-17	2 בתים (Word): מספר גזרות בטבלת FAT.
* 18-19	2 בתים (Word): מספר גזרות במסלול (Sectors/Track).
* 1A-1B	2 בתים (Word): מספר משטחי מגנוט בדיסק (Heads).
* 1C-1F	4 בתים (Double Word): מספר גזרות נסתרות (Hidden Sectors).
* 20-23	4 בתים (Double Word): מספר גזרות לוגיות (Logical Sectors).

(*) ערכים אלה אינם בשימוש של מערכת ההפעלה. הם משמשים לצורך גישה למצעי האחסון המגנטיים על ידי תכניות הפעלה להתקנים (Device Drivers).

ניתן לראות פרמטרים אלה באמצעות תכניות שירות כדוגמת Norton, או PC-Tools.



מבנה רשומת BOOT

5.4.2.3 טבלת מיקום קבצים — FAT (File Allocation Table)

ציון מספר לכל יחידות ההקצאה (Allocation Units) בדיסק/דיסקט הופך את הדיסק למשטח מגנטי לוגי רציף. אם הקבצים היו נכתבים לדיסק/דיסקט בזה אחר זה בצורה רציפה ולא היינו מוחקים קבצים מהדיסק/דיסקט, לא היתה נוצרת כל בעיה בהקצאת שטח לקבצים חדשים. הקובץ הראשון היה נמצא תמיד בתחילת הדיסק/דיסקט, קובץ שני אחריו וכו'. מכיון שניהול הקבצים במערכת נעשה בצורה דינמית שבה מוסיפים ומוחקים קבצים, תוכן הדיסק/ דיסקט משתנה במהלך העבודה.

מחיקת קבצים יוצרת "חורים" ברצף יחידות ההקצאה, אשר מנוצלים לצורך כתיבה של קבצים חדשים. על מנת לעקוב אחר המקומות המוקצים והפנויים בדיסק קיימת טבלת FAT. לכל יחידת הקצאה יש כניסה בטבלה, אשר מציינת אם היא פנויה או מוקצית. אם היא מוקצית, יש ציון היכן לחפש את המשך הקובץ. גודל הכניסה המתארת את מצב יחידת ההקצאה תלוי בגודל הדיסק/ דיסקט ובגודל יחידת ההקצאה. כניסה בטבלת FAT יכולה לציין את המצבים הבאים:

* יחידת הקצאה פנויה. כאשר צריך להקצות מקום לקובץ, ניתן להשתמש ביחידה זאת. הערך 0 מציין שיחידת ההקצאה פנויה.

* יחידת הקצאה אחרונה בשרשרת. זאת יחידת ההקצאה האחרונה המוקצית לקובץ. הערך המספרי המציין יחידת הקצאה אחרונה הוא FF8-FFFHex (או FFF8-FFFFHex, אם גודל כניסה בטבלת FAT הוא 16 סיביות).

* מספר יחידת ההקצאה הבאה. ערך זה מציין את המספר של יחידת ההקצאה הבאה המוקצית לקובץ. הערך המספרי של יחידת ההקצאה הבאה יכול להיות מ-1 ועד מספר יחידת ההקצאה האחרונה בדיסק/דיסקט.

* יחידת הקצאה לא תקינה. ציון כי יחידת ההקצאה לא תקינה ולא ניתן להשתמש בה. בתהליך Format מציינים את כל האיזורים שנתגלו כלא תקינים על מנת שמערכת ההפעלה לא תכתוב בהם נתונים. הערך המספרי המציין יחידה לא תקינה הוא FF0-FF7Hex (או FFF0-FFF7Hex, אם גודל כניסה בטבלת FAT הוא 16 סיביות).

הטבלה אינה מציינת מי היא יחידת ההקצאה הראשונה המוקצית לקובץ. מספר זה נמצא בספריה (Directory).

רישום ההקצאה ב-FAT

טבלת FAT מתארת את ההקצאה של Allocation Units לקבצים. לכל יחידת הקצאה בדיסקט (או בדיסק שגודלו מתחת ל-2MByte) מוקצים ב-FAT 12 סיביות (בית וחצי), שבעזרתן ניתן למנות 4096 יחידות הקצאה בנות גזרה אחת ועל כן בדיסקט בעל קיבולת של 1.44MByte ישנן:

2 צדדים x 80 מסלולים x 18 גזרות במסלול = 2880 יחידות הקצאה שונות.

```
                פקודת קפיצה          שם יצרן תוכנה          BPB(קו מתחת)
0000  EB 3C 90 4D 53 44 4F 53 34 2E 30 00 02 02 01 00   δ<ɔMSDOS4.0.....
0010  02 70 00 D0 02 FD 02 00 09 00 02 00 00 00 00 00   .p...².........
0020  00 00 00 00 00 00 29 D9 10 0B 0F 4E 4F 20 4E 41   ......) ...NO NA
0030  4D 45 20 20 20 20 46 41 54 31 32 20 20 20 FA 33   ME    FAT12   ·3
0040  C0 8E D0 BC 00 7C 16 07 BB 78 00 36 C5 37 1E 56   └מ.σ.¦..Σx.6┼7V
0050  16 53 BF 3E 7C B9 0B 00 FC F3 A4 06 1F C6 45 FE   .S┐>¦Γ..ⁿ≤ñ.
0060  0F 8B 0E 18 7C 88 4D F9 89 47 02 C7 07 3E 7C FB   .ט¦..ɔM˙›G.├.>¦√
0070  CD 13 72 7C 33 C0 39 06 13 7C 74 08 8B 0E 13 7C   .r¦3└9..¦t.כ..¦
0080  89 0E 20 7C A0 10 7C F7 26 16 7C 03 06 1C 7C 13   ›. ¦á.¦≈&.¦..¦.
0090  16 1E 7C 03 06 0E 7C 83 D2 00 A3 50 7C 89 16 52   .¦...¦ד..úP¦›.R
00A0  7C A3 49 7C 89 16 4B 7C B8 20 00 F7 26 11 7C 8B   ¦úI¦›.K¦β .≈&.¦ס
00B0  1E 0B 7C 03 C3 48 F7 F3 01 06 49 7C 83 16 4B 7C   2.¦.├H≈≤..I¦ד.K¦
00C0  00 BB 00 05 8B 16 52 7C A1 50 7C E8 87 00 72 20   .Σ..ס.R¦íP¦Φח.r
00D0  B0 01 E8 A1 00 72 19 8B FB B9 0B 00 BE DB 7D F3   ░.Φí.r.ס√Γ..τ }≤
00E0  A6 75 0D 8D 7F 20 BE E6 7D B9 0B 00 F3 A6 74 18   ªu.ם τ.}Γ..≤ªt.
00F0  BE 93 7D E8 51 00 32 E4 CD 16 5E 1F 8F 04 8F 44   τף}ΦQ.2..^
0100  02 CD 19 58 58 58 EB E8 BB 00 07 B9 03 00 A1 49   ..XXXδΦΣ..Γ..íI
0110  7C 8B 16 4B 7C 50 52 51 E8 3A 00 72 E6 B0 01 E8   ¦ס.K¦PRQΦ:.r.░.Φ
0120  54 00 59 5A 58 72 C9 05 01 00 83 D2 00 03 1E 0B   T.YZXr...ד....
0130  7C E2 E2 8A 2E 15 7C 8A 16 24 7C 8B 1E 49 7C A1   ¦..ך¦..ך.$¦סI¦í
0140  4B 7C EA 00 00 70 00 AC 0A C0 74 29 B4 0E BB 07   K¦Ω..p.ה.└t)┤.Σ.
0150  00 CD 10 EB F2 3B 16 18 7C 73 19 F7 36 18 7C FE   ..δ≥;..¦s.≈6.¦■
0160  C2 88 16 4F 7C 33 D2 F7 36 1A 7C 88 16 25 7C A3   ┬ט.O¦3.≈6.¦ט.%¦ú
0170  4D 7C F8 C3 F9 C3 B4 02 8B 16 4D 7C B1 06 D2 E6   M¦˚├˙├┤.ס.M¦▒...
0180  0A 36 4F 7C 8B CA 86 E9 8A 16 24 7C 8A 36 25 7C   .6O¦ך θזס.$¦ך6%¦
0190  CD 13 C3 0D 0A 4E 6F 6E 2D 53 79 73 74 65 6D 20   .├..Non-System 
01A0  64 69 73 6B 20 6F 72 20 64 69 73 6B 20 65 72 72   disk or disk err
01B0  6F 72 0D 0A 52 65 70 6C 61 63 65 20 61 6E 64 20   or..Replace and 
01C0  70 72 65 73 73 20 61 6E 79 20 6B 65 79 20 77 68   press any key wh
01D0  65 6E 20 72 65 61 64 79 0D 0A 00 49 4F 20 20 20   en ready...IO   
01E0  20 20 20 53 59 53 4D 53 44 4F 53 20 20 20 53 59      SYSMSDOS   SY
01F0  53 00 00 00 00 00 00 00 00 00 00 00 00 00 55 AA   S.............UĀ
                 תכנית הטעינה (מודגשת)          סימון
                                                לגזרת
                                                BOOT
```

מבנה גזרת BOOT

בדיסק קשיח מקצים ב-FAT לכל יחידת הקצאה כניסה אחת ברוחב 16 סיביות (2 בתים), שבעזרתן ניתן למנות 65536 יחידות בנות 512 בתים, או להגדיר דיסק בקיבולת של 32MByte. כניסה של 16 סיביות מנעה ממערכת ההפעלה בגירסאות 3.3 ומטה אפשרות לפעול עם דיסקים בקיבולת שמעל 32MByte כדיסק יחיד וחייבה חלוקתם למספר דיסקים לוגיים בעזרת תכנית FDISK של מערכת ההפעלה, או כלי תוכנה אחרים. במערכת ההפעלה מגירסה

4.0 ואילך התגברו על בעיה זו באמצעות הקצאת כניסה ברוחב 32 סיביות למספור יחידות ההקצאה, אם צריך לעשות זאת מפאת גודל הדיסק.

מערכת ההפעלה רושמת את פירוט הקצאת המקום בדיסק/דיסקט בשתי טבלאות FAT במקביל, לצורך אמינות ובדיקה של נתונים, פרט למקרים שבהם הנתונים אינם נרשמים על משטח מגנטי, כי הדיסק מוגדר בזיכרון RAM של המחשב. שתי כניסות ראשונות בטבלת FAT מכילות את הנתונים הדרושים לצורך טיפול בדיסקט וב-FAT: סוג הדיסקט ורוחב כניסה בטבלת FAT (12 סיביות, 16 סיביות, או 32 סיביות מגירסה 4.0).

מן הכניסה השלישית ואילך נמצא בטבלת FAT מיפוי מדויק של יחידות ההקצאה השונות המשמשות לשמירת נתוני הקבצים. כלומר, אם במקום ה-18 בטבלת FAT קיים הערך 0, פירוש הדבר שיחידה 18 פנויה וניתן לכתוב בה נתונים.

ההקצאה לקבצים נעשית על ידי מערכת ההפעלה, אשר "מפשירה" יחידת הקצאה אחת בכל פעם שנחוץ מקום נוסף לקובץ. היחידה הראשונה הפנויה בטבלת FAT היא זו שתוקצה. אלגוריתם הקצאה זה אינו מחייב שההקצאה לקובץ תיעשה באמצעות יחידות הקצאה עוקבות ועל כן ייתכן שהקובץ לא יימצא ברצף על גבי הדיסק/דיסקט. האלגוריתם יעיל ביותר לניצול שטחים שנתפנו לאחר מחיקת קבצים, אך הוא איננו יעיל מבחינת זמן הגישה לנתונים. שטחים שהתפנו יוקצו שוב בבקשת ההקצאה הבאה ולכן נוצרת אי רציפות בשטח הקובץ.

אלגוריתם ההקצאה

אלגוריתם ההקצאה יכול ליצור מצב, שבו הנתונים של קובץ אחד מפוזרים על גבי מספר מסלולים הרחוקים זה מזה, זהו **קיטוע הקובץ** (File Fragmentation). במצב זה, תידרשנה תנועות רבות של הראש הקורא, לצורך קריאת הקובץ, שמשמעותן הארכת משך הקריאה והורדת הביצועים של המחשב. יש לדבר משמעות בעיקר בקובצי נתונים גדולים על גבי דיסק קשיח. אלגוריתם הקצאה יעיל יותר מבחינת תנועות זרוע של הראש הקורא, הוא הקצאה בבת אחת של כל יחידות ההקצאה הנדרשות לקובץ באופן שכולן עוקבות זו לזו.

אלגוריתם המקצה ברצף את כל יחידות ההקצאה הדרושות לקובץ, מחייב חיפוש שטחים פנויים וצורך יותר זמן. כמו כן, לא כל בקשות ההקצאה לקבצים נעשות בצורה רציפה. קבצים שגדלים או קטנים באופן שוטף מחייבים הקצאה של יחידות בודדות נוספות. אלגוריתם הקצאה רציף יגרום לשטחים מתים בדיסק/דיסקט שאינם מספיקים לאחסון קבצים גדולים, כלומר לקיטוע (Fragmentation), אם יחידות ההקצאה אינן עוקבות.

כדי להקטין את פיזור הקצאת היחידות על ידי אלגוריתם ההקצאה של מערכת ההפעלה צריך לארגן מדי פעם מחדש את הקבצים. כלומר, שצריך לכתוב אותם מחדש בצורה רציפה על גבי הדיסק/דיסקט. לשם כך ניתן להשתמש בתכניות שירות כמו Pc-Tools6 או Norton 5 (ראה פרק על אחזקת דיסקים קשיחים). לאחר פעולה זו יקטנו זמני העיבוד של הקבצים – לעתים באופן משמעותי.

יחידות הקצאה לקובץ משורשרות זו לזו. נציג זאת בדוגמה של קטע מסוים מה-FAT. המספרים מעל הריבועים מציינים מספר יחידת הקצאה והמספרים בתוך הריבועים מציינים את יחידת ההקצאה הבאה, או סוף הקצאה.

מקום בטבלה	18	17	16	15	14	13	12	11	10	
תוכן הטבלה	0	FFF	17	16	FFF	15	FFF	13	11	

דוגמה לקטע מטבלת FAT

בטבלה הנתונה, לקובץ אשר מספר יחידת ההקצאה הראשונה עבורו הוא 10 יש 6 יחידות הקצאה שמספרן: 10, 11, 13, 15, 16, 17. עבור כל יחידת הקצאה רשום בטבלה המספר הסידורי של יחידת ההקצאה הבאה. מספר 0 מציין יחידת הקצאה פנויה לצורכי הקצאה, מספר בין FF8 ל-FFF מציין סוף שרשרת הקצאה, מספר בין FF0 ל-FF7 מציין יחידת הקצאה לא תקינה. כך, בדוגמה זו, היחידה הראשונה שהוקצתה לקובץ מסוים היא 10 ואחריה 11, 13 ואח"כ 15 וכן הלאה עד יחידה 17 שהיא האחרונה ברשימת ההקצאה לקובץ זה. לקבצים המתחילים ביחידות הקצאה שמספרן 12 ו-14 יש יחידת הקצאה אחת לכל קובץ. יחידת הקצאה 18 פנויה וניתן להקצותה לקבצים חדשים. המשתמש או המתכנת, אינו מטפל בטבלת הקצאת קבצים, מכיון שכל בקשות ההקצאה והשחרור נעשות על ידי מערכת ההפעלה DOS שמנהלת את טבלת FAT.

ניתן להציג פרמטרים אלה בדוח בעזרת תכניות שירות כדוגמת Norton או PC-Tools.

5.4.3 ספריית השורש (Root Directory)

בזמן תיבנות הדיסק או הדיסקט, תכנית FORMAT בונה ומקצה מקום לספריית השורש. מיקום ספריית השורש הוא לאחר שני עותקי טבלת מיקום הקבצים (FAT) והוא משתנה בהתאם לסוגי וקיבולות הדיסקים/דיסקטים השונים. ספריית השורש היא טבלה קבועה בגודלה, שכל כניסה בה הינה בגודל של 32 בתים. היא מתארת קובץ יחיד, או תת-ספריה הנמצאים על הדיסק/הדיסקט. גודל ספריית השורש קבוע (בהתאם לסוג הדיסק/הדיסקט) ועל כן היא יכולה להכיל מספר מוגבל של כניסות. לעומת זאת, **תת-ספריות** (Sub Directories) הן טבלאות דינמיות המוחזקות כקבצים לכל דבר, ועל כן הן יכולות להכיל רישום של מספר לא מוגבל של קבצים, או תת-ספריות נוספות.

פרטי קובץ הנשמרים בספריה

בתים 0-7 - **שם הקובץ:** 8 בתים מכילים את שם הקובץ ב-ASCII ללא סיומת. בית 0 עשוי להכיל ערכים בינאריים שאינם מותרים בשם קובץ ומשמעותם "תיאור מצב הקובץ". רשימת הקודים:

00 Hex - לא קיים קובץ בכניסה זאת וגם בעבר לא היה בה קובץ.

05 Hex - ציון שהאות הראשונה של שם הקובץ היא באמת E5 Hex ואין זה קובץ שנמחק (ראה קוד E5).

2E Hex - ערך זה (התו ".") בטבלת ASCII) מציין כניסה של תת-ספריה, בתנאי שגם הבית השני של השם מכיל את התו ".". מספר יחידות ההקצאה מצביע על מיקום הספריה של האבא. אלה הקבצים ששמם "." ו-".." אשר מוצגים על המסך כאשר מקישים פקודת DIR בתת-ספריה.

E5 Hex - קובץ היה רשום בכניסה זו, אך נמחק. סימון קובץ כמבוטל נעשה על ידי ביטול האות הראשונה בשמו. על כן, צריך להקיש את האות הראשונה של שם הקובץ בפעולת שחזור Undelete בתכניות כמו Norton או PC-Tools.

כל ערך אחר מציין קובץ שקיים בדיסקט/דיסק.

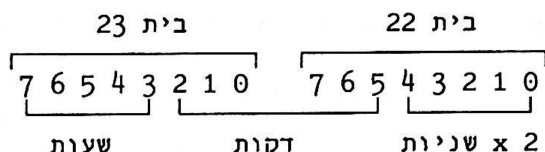
בתים 8-10 - **הסיומת** של הקובץ (Extension): 3 תווים.

בית 11 - **התכונות** של הקובץ (Attribute). כל סיבית בבית זה מציינת תכונה שונה, והיא יכולה לקבל ערך 0 או 1 על פי הפירוט הבא:

מספר סיבית	
0	ערך 1 - קובץ שניתן לקריאה בלבד
1	ערך 1 - קובץ נסתר (Hidden)
2	ערך 1 - קובץ מערכת (System)
3	ערך 1 - 11 הבתים הראשונים הם שם הדיסקט ושאר הבתים אינם מכילים נתונים (Volume Lable)
4	ערך 1 - הכניסה היא עבור תת-ספריה. כלומר, הקובץ שכניסת טבלה זו מצביעה עליה אינו קובץ, אלא תת-ספריה
5	ערך 1 - סיבית הארכיב (Archive BIT). הסיבית משמשת כמנגנון בקרה בתכניות XCOPY, BACKUP כדי למנוע העתקת קבצים לא רצויים
6	ערך 0 קבוע
7	ערך 0 קבוע

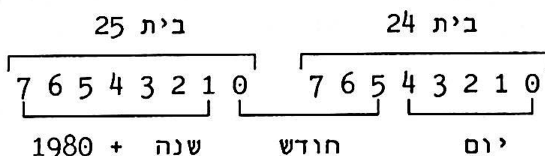
בתים 12-21 - שמורים למערכת ההפעלה.

בתים 22-23 - הזמן הקלנדרי שבו נוצר הקובץ, או עודכן לאחרונה:



הערה: זוהי הצורה התקנית של מערכת ההפעלה DOS לרישום זמן (שים לב שהשניות תמיד זוגיות).

בתים 24-25 - התאריך שבו נוצר הקובץ או עודכן לאחרונה:



בתים 26-27 - המספר של יחידת ההקצאה הראשונה עבור הקובץ. עבור הקובץ שתואר בדוגמת FAT למעלה, מכילה כניסה זו את המספר 10 כדי לציין את המספר של יחידת ההקצאה הראשונה של הקובץ.

בתים 28-31 - גודל הקובץ בבתים.

תכנית דוגמה לטיפול בנתוני קבצים ניתן למצוא בנספח תכניות הדוגמה.

סוג דיסקט	גזרות במסלול	גזרות ב-FAT	מספר גזרות בספריית שורש	מספר מירבי של קבצים	מספר גזרות ב-.Allo. U	אפיון ב-BPB
5 1/4	9	2	7	112	2	FD
3 1/2	9	3	7	112	2	F9
5 1/4	15	7	14	224	1	F9
3 1/2	18	9	14	224	1	F0

מספר קבצים מירבי בספריית השורש

5.4.4 איזור הנתונים בדיסק/דיסקט (Data Area)

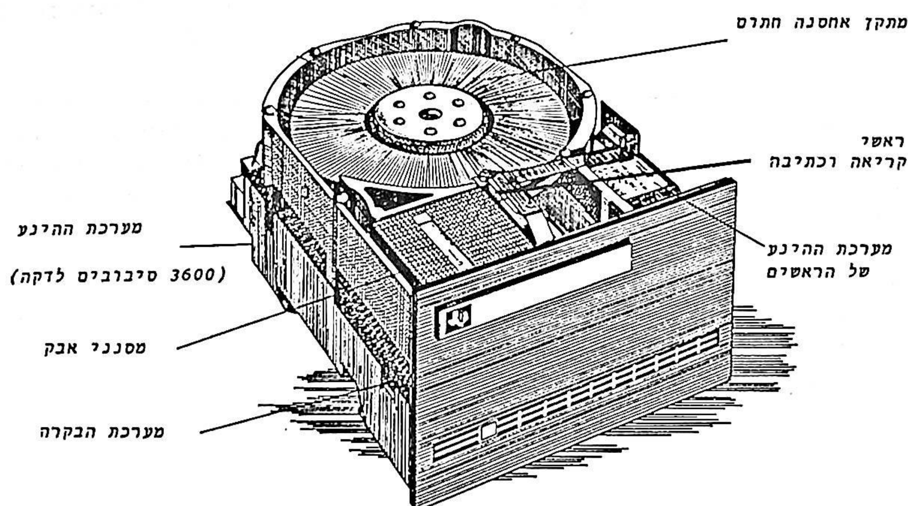
באיזור הנתונים שעל הדיסקט/דיסק שומרת מערכת ההפעלה את תוכן הקבצים. יחידת ההקצאה הבסיסית לקובץ היא יחידת הקצאה אחת (Allocation Unit) ללא תלות בגודלו, ואפילו אם הקובץ קטן מיחידת הקצאה אחת. אם יחידת ההקצאה היא גזרה אחת, המושגים גזרה ויחידת הקצאה זהים. כאשר הקובץ גדל ומתרחב מקצים לו את היחידה הפנויה הראשונה על פי טבלת FAT.

איזור הנתונים אינו כולל את כל שטח הדיסק/דיסקט, מכיון שחלק מקיבולת זו מוקצית למערכת ההפעלה (BOOT ,FAT ,Directory). מערכת ההפעלה מקצה

שטחי אחסון בשיטת "ראשון פנוי ראשון מוקצה". אלגוריתם זה הוא הפשוט ביותר ומבטיח שאם קיימות יחידות הקצאה עוקבות פנויות, תהיה תנועה מינימלית של זרוע הראש הקורא. מצב אופטימלי זה הופך בלתי אפשרי ככל שהמשטח מתמלא וישנן כל הזמן פעולות הקצאה ומחיקה.

סוג דיסקט	גזרות במסלול	מסלולים	קיבולת כוללת	קיבולת לנתונים
5 1/4	9	40	368,640	362,496
3 1/2	9	80	737,280	730,112
5 1/4	15	80	1,228,800	1,213,952
3 1/2	18	80	1,474,560	1,457,664

קיבולת איזור הנתונים בדיסקטים השונים



דיסק וינצ'סטר

5.4.5 הדיסק הקשיח

5.4.5.1 חלוקה למחיצות

תהליך Pre-Format לדיסק הקשיח אינו מספיק על מנת להכין אותו לפעולה תחת מערכת ההפעלה DOS ולהקצות בו את שטחי העבודה הדרושים. כדי לעשות זאת, יש לבצע את החלוקה **למחיצות** (Partitions). חלוקה למחיצות היא הגדרה של הדיסק הפיסי כיחידה אחת, או חלוקה לוגית שלו למספר יחידות משנה, אשר כל אחת מהן מהווה דיסק לוגי וכוללת מספר צילינדרים עוקבים של הדיסק. מספר סיבות גרמו את הצורך לחלק את הדיסק למחיצות:

* עד גירסה 4.0 של מערכת ההפעלה DOS לא ניתן היה להשתמש **במחיצת DOS** (DOS Partition) אשר גדולה מ-32MBytes. הדבר נבע מכך, שכל כניסה של FAT היתה ברוחב של 16 סיביות, אשר יכולה לייצג מספר שערכו המקסימלי 65536. מספר זה של גזרות, אשר כל אחת מהן היא בת 512 בתים, אינו מאפשר לעבור את מחסום 32MByte. מסיבה זו צריך לחלק את נפח הדיסק שקיבולתו גדולה מ-32MBytes למספר דיסקים לוגיים (Partitios). בגירסה 4.0 ומעלה של מערכת ההפעלה נפתרה הבעיה על ידי כך, שכל כניסה בטבלת FAT הינה ברוחב של 32 סיביות, אשר מאפשרת להגדיר מחיצות גדולות.

* חלוקה של הדיסק למספר מחיצות לוגיות גורמת לכך שתנועת הראש הקורא למסלול המבוקש הינה מהירה יותר, מכיון שהוא צריך לעבור על (Seek) פני מספר קטן יותר של מסלולים כדי להגיע למסלול המבוקש. משך הבאת הראש הקורא למסלול המבוקש הוא פרק הזמן הארוך ביותר במהלך הקריאה/ כתיבה ועל כן קיצורו יתרום לכך שהגישה תתבצע במהירות גבוהה יותר וביצועי המחשב יעלו.

החלוקה למחיצות נרשמת בגזרת BOOT של הדיסק הקשיח, אשר נמצאת בצד 0, מסלול 0, גזרה 1 של הדיסק. **בטבלה המחיצות** (Partition Table) מפורטת החלוקה הלוגית של הדיסק למחיצות. עבור כל מחיצה נמצא את המסילה שבו היא מתחילה, את המסילה האחרונה שלה ואת סוג המחיצה. בתהליך BOOT של המחשב המתבצע דרך הדיסק הקשיח, קוראת תכנית BIOS את גזרת BOOT מהדיסק, בוחנת את טבלת המחיצות וטוענת לזיכרון את הגזרה הראשונה שבמחיצה המסומנת כ"מחיצת BOOT" של מערכת ההפעלה DOS.

לאחר הטעינה מתבצעת תכנית BOOT, אשר ממשיכה בתהליך וטוענת את מערכת ההפעלה DOS לזיכרון המחשב. רק **מחיצה אחת** מכל המחיצות בדיסק פיסי יכולה להיות מסומנת כמחיצת BOOT.

החלוקה של הדיסק הקשיח למספר מחיצות נעשית באמצעות תכנית FDISK של מערכת ההפעלה DOS. היא מאפשרת לחלק את הדיסק הקשיח למספר רב של דיסקים לוגיים, אך יש להתחשב במגבלות מערכת ההפעלה DOS ובתועלת או הנזק שיצמחו מחלוקה למספר רב של דיסקים לוגיים. ראה הסברים על דיסקים לוגיים מסוגים שונים בסעיף 3.3.6.

מערכת ההפעלה DOS מכירה בשמות של עד 5 כוננים (A-E) כברירת מחדל. כל מחיצה או דיסק לוגי מוסיפה אות לציון של השם למערכת ההפעלה והדבר משמש לזיהוי בעת פניה אליהם. החלוקה של הדיסק למחיצות באמצעות תכנית FDISK מאפשרת למערכת ההפעלה להגדיר את הכונן האחרון שאליו ניתן לגשת,

על פי מספר הכוננים הלוגיים המוגדרים. הכונן האחרון מוגדר על ידי אות, כמו למשל :F). דיסקים לוגיים ניתן להגדיר מ-:A עד :Z, כלומר עד 26 דיסקים לוגים. במקרה כזה, האות האחרונה תהיה :Z. למרות האפשרות להגדיר דיסקים לוגיים רבים, אין אנו ממליצים לעשות זאת. גם במערכות גדולות לא יעיל להגדיר יותר מ-4 עד 6 דיסקים לוגיים.

בשוק קיימות תוכנות כדוגמת Disk Manager, המאפשרות לחלק את הדיסק למחיצות באמצעות תכנית חיצונית למערכת ההפעלה ולעקוף את DOS. חלוקה למחיצות בעזרת תכניות חיצוניות מחייבת את המשתמש להגדיר בעצמו באמצעות הפקודה LASTDRIVE בקובץ CONFIG.SYS, את שם הדיסק הלוגי האחרון שאליו ניתן לגשת. בנוסף, חלוקת הדיסק למחיצות על ידי תוכנית חיצונית מחייבת להשתמש בתכנית Device driver מתאימה, המכירה את החלוקה הזאת. גם תכנית זו תוטען על ידי המשתמש באמצעות פקודה מתאימה בקובץ CONFIG.SYS. אם המשמש לא יגדיר פקודה לטעינת תכנית Device driver שלו לטיפול בדיסק, מערכת ההפעלה תכיר ותדע להשתמש בדיסק הלוגי הראשון בלבד ולא תכיר את שאר הדיסקים הלוגיים. לפיכך, אנו ממליצים ששימוש בכלים מחוץ למערכת ההפעלה ייעשה רק כאשר הדבר הכרחי, ורק על ידי אנשים מיומנים.

בגזרת BOOT הראשית של הדיסק (צד 0, מסלול 0, גזרה 1) מפורטת טבלת המחיצות (Partition table). בטבלה זו ניתן להגדיר עד 4 מחיצות שונות הנפרדות זו מזו. אפשר להגדיר מחיצות רבות יותר באמצעות המחיצה המורחבת, כפי שיוסבר בהמשך.

סוגי המחיצות:

DOS-PRIMARY – **מחיצת** DOS **ראשית** (FAT של 12/16/32 סיביות, או בהתאם לגודל המחיצה).

DOS-EXTENDED – **מחיצה מורחבת** המאפשרת לכלול בתוכה מספר דיסקים לוגיים. סוג מחיצה זה מאפשר חלוקה של דיסק גדול מאוד למספר דיסקים לוגיים.

NON-DOS – **מחיצה שאינה שייכת ל-DOS**. מחיצה כזו יכולה לשמש לצורכי מערכת הפעלה אחרת שתפעל על אותו דיסק, או שהיא בשימוש של תכניות כמו Disk Manager, אשר מחלקות את הדיסק בצורה שונה ממערכת DOS ומספקות Device driver לצורך הפעלתו.

עבור כל אחת מארבע המחיצות מוקצים בטבלת המחיצות **16 בתים** המגדירים ומפרטים את המחיצה. הטבלה נמצאת החל מכתובת 1BE Hex בגזרת BOOT הראשית ומתייחסת למחיצה הראשונה בדיסק. להלן תיאור הטבלה:

	1BE	1BF	1C0	1C1
התחלה	BOOT Indicator	Head	Sector	Cylinder

	1C2	1C3	1C4	1C5
סיום	System Indicator	Head	Sector	Cylinder

	1C6 1C7	1C8 1C9
גזרות יחסיות	מלה תחתונה	מלה עליונה

	1CA 1CB	1CC 1CD
מספר גזרות	מלה תחתונה	מלה עליונה

קטע מטבלת מחיצות - מחיצה ראשונה

הסבר:

BOOT Indicator – ערך 80Hex בבית זה מסמן כי בתהליך BOOT זוהי המחיצה שיש לבצע דרכה BOOT. ערך 0 מציין כי דרך מחיצה זו לא ניתן לבצע BOOT. רק מחיצה אחת בדיסק יכולה להיות מחיצת BOOT.

System Indicator – סוג המחיצה, על פי הפירוט הבא:
ערך 00Hex – מחיצה שאינה מחיצת DOS.
ערך 01Hex – מחיצת DOS ראשונית (Primary) עם כניסה ב-FAT ברוחב של 12 סיביות.
ערך 04Hex – מחיצת DOS ראשונית (Primary) עם כניסה ב-FAT ברוחב של 16 סיביות.
ערך 05Hex – מחיצת DOS מורחבת (Extended).

התחלה – 3 ערכים מציינים את המסלול, הצד והגזרה שבהם מתחילה המחיצה.

Head – מספר הצד.

Sector – מספר גזרת התחלה במסלול.

Cylinder – מספר המסלול של התחלת המחיצה בדיסק. 2 סיביות עליונות של מספר זה נמצאות ב-2 סיביות עליונות של הבית המציין את מספר גזרת התחלה.

סיום – 3 ערכים מציינים את הגזרה, המסלול והצד שבהם מסתיימת המחיצה (כמו בהתחלת מחיצה).

גזרות יחסיות – מספר בן 4 בתים, אשר מכיל את מספר הגזרות שהוקצו לפני מחיצה זו. הספירה מתחילה מצד 0, מסלול 0, גזרה 1 וממשיכה באותו צד, עוברת לצד הבא ולאחר מכן למסלול הבא. כלומר הספירה נעשית באותו אופן שהיא נעשית עבור מספרי יחידות ההקצאה.

מספר גזרות – מספר הגזרות שמוקצות למחיצה.

```
0000  FA 2B C0 8E D0 8E C0 8E D8 B8 00 7C 8B E0 FB 8B  ·+└כ√.כ¦.β┼מ└מ.מ
0010  F0 BF 00 7E FC B9 00 01 F3 A5 E9 00 02 B9 10 00  ≡┐.~ⁿΓ..≤Ñθ..Γ..
0020  8B 36 85 7E F6 04 80 75 08 83 EE 10 E2 F6 EB 37  כ6א.+~וu.דε..+δ7
0030  90 BF BE 07 57 B9 08 00 F3 A5 5E BB 00 7C 8B 14  נ┐ד.WΓ..≤Ñ^Σ.¦כ.
0040  8B 4C 02 BD 05 00 B8 01 02 CD 13 73 09 2B C0 CD  כL.µ..β...s.+└
0050  13 4D 74 19 EB F0 BE FE 7D AD 3D 55 AA 75 14 BE  .Mt.δ≡τ■}¡=UÃu.τ
0060  BE 07 EA 00 7C 00 00 8B 36 87 7E EB 0A 8B 36 89  τ.Ω.¦..כ6ח.δ~n6י
0070  7E EB 04 8B 36 8B 7E AC 0A C0 74 FE BB 07 00 B4  ~δ.כ6כ~h.└t■Σ..┤
0080  0E CD 10 EB F2 EE 7F 8D 7E A7 7E C8 7E 0D 0A 49  ..δ≥ε ם~º~╚~..I
0090  6E 76 61 6C 69 64 20 50 61 72 74 69 74 69 6F 6E  nvalid Partition
00A0  20 54 61 62 6C 65 00 0D 0A 45 72 72 6F 72 20 4C   Table...Error L
00B0  6F 61 64 69 6E 67 20 4F 70 65 72 61 74 69 6E 67  oading Operating
00C0  20 53 79 73 74 65 6D 00 0D 0A 4D 69 73 73 69 6E   System...Missin
00D0  67 20 4F 70 65 72 61 74 69 6E 67 20 53 79 73 74  g Operating Syst
00E0  65 6D 00 00 00 00 00 00 00 00 00 00 00 00 00 00  em..............
00F0  00 00 00 00 00 00 00 00 00 00 00 AA 55 00 00  ...........ÃU..
0100  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0110  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0120  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0130  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0140  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0150  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0160  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0170  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0180  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0190  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
01A0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
01B0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 80 01  ..............א.
01C0  01 00 06 03 11 3B 11 00 00 00 DF 0F 00 00 00 00  .....;....■.....
01D0  01 3C 51 03 91 65 F0 0F 00 00 28 93 00 00 00 00  .<Q.סe≡...(ף....
01E0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
01F0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 55 AA  ..............UÃ
```

טבלת המחיצות (מודגשת)

סימון לגזרת BOOT קיימת

גזרת BOOT ראשית בדיסק קשיח וטבלת המחיצות

בגזרת BOOT הראשית בדיסק ישנן 4 מחיצות, שבכל אחת מהן 16 בתים רצופים אשר מתארים את 4 המחיצות השונות האפשריות. תיאור המחיצה הראשונה מתחיל בכתובת 1BEHex, תיאור המחיצה השנייה – בכתובת 1CEHex, תיאור המחיצה השלישית – ב-1DEHex ותיאור המחיצה הרביעית – ב-1EEHex.

ניתן להציג פרמטרים אלה בדוח ערוך בעזרת תכניות שירות כדוגמת Norton או PC-Tools.

בדוגמה שבשרטוט מוצגת טבלת מחיצות לדיסק קשיח, שגודלו הפיסי 21MByte. הדיסק מחולק לשתי מחיצות שהוכנו באמצעות תכנית Disk Manager:

* מחיצת DOS שניתן לבצע דרכה BOOT (ערך 80Hex) בגודל 2MByte.
* מחיצה שאיננה מחיצת DOS, בגודל 19MByte.
* שתי מחיצות נוספות שאינן בשימוש וכל הערכים בהן 0.

מחיצה מורחבת (Extended Partition)

מחיצה מורחבת מאפשרת חלוקה של דיסקים גדולים למספר דיסקים לוגיים, אשר נוחים יותר לניהול על ידי המשתמש. בדיסק המשמש ל-BOOT ניתן ליצור מחיצה מורחבת רק לאחר שיוצרים בו מחיצת DOS ראשונית. בדיסק שאינו משמש ל-BOOT (דיסק D למשל) ניתן ליצור מחיצה מורחבת, גם ללא מחיצת DOS ראשונית. מחיצה מורחבת אינה מאפשרת לבצע BOOT של המערכת.

מגדירים מחיצה מורחבת באמצעות תכנית FDISK של מערכת ההפעלה. היא מתחילה ומסתיימת תמיד בכפולות של צילינדרים שלמים. "גזרת BOOT", שהינה הגזרה הראשונה במחיצה זו (גזרה 1 של מסלול ראשון, צד 0 במחיצה המורחבת) דומה ותואמת לגזרת BOOT הראשית של הדיסק ומשמשת לטבלת המחיצות הפנימית. אף על פי שבטבלת המחיצות של המחיצה המורחבת יש גם כן מקום ל-4 כניסות, ניתן להגדיר עבור כל מחיצה כזאת רק כניסה אחת עבור כל דיסק לוגי וכניסה נוספת, המשמשת כמצביע עבור הדיסק הלוגי הבא. השרשור של הדיסקים הלוגיים יוצר אוסף של דיסקים לוגיים כשהקישור (Link) ביניהם נעשה באמצעות גזרת BOOT הנמצאת בתחילת כל דיסק לוגי.

דיסקים לוגיים שהוגדרו באמצעות תכנית FDISK של מערכת ההפעלה DOS מוכרים על ידה באופן אוטומטי והמשתמש לא צריך להגדיר פרטים אלה בקובץ CONFIG.SYS.

5.4.5.2 תהליך FORMAT בדיסק קשיח

לאחר חלוקת הדיסק למחיצות באמצעות תכנית FDISK, יש לערוך אותו לשימוש באמצעות פקודת מערכת ההפעלה FORMAT. עריכה זו יש לבצע בנפרד לכל דיסק לוגי אשר מוגדר בדיסק הפיסי. אם, למשל, דיסק פיסי C מחולק לשני דיסקים לוגיים C ו-D, יש לבצע עריכה לכל אחד מהם. כזכור, תהליך FORMAT בדיסק קשיח עורך את האיזורים השונים המשמשים את מערכת ההפעלה (גזרת BOOT, FAT, Root Directory), אך לא פוגע באיזור הנתונים.

תהליך FORMAT בדיסק קשיח נבדל מהתהליך המתקיים בכונן דיסקטים בכך שהוא **אינו** מוחק את הנתונים שבדיסק על ידי כתיבה מחדש. על כן, **אם מבצעים בטעות FORMAT לדיסק קשיח ניתן לשחזר את הקבצים שהיו בו**

באמצעות תכניות ייעודיות לפעולת שחזור, כמו Norton או PC-Tools.
פעולת FORMAT בדיסקט היא פעולה **בלתי הפיכה**. בגירסה 5 ומעלה של DOS ניתן לשחזר את הדיסקט אם הפעולה נעשתה בעזרת הפקודה הסטנדרטית.

5.4.5.3 הכנת דיסק קשיח לשימוש

על מנת להכין את הדיסק הקשיח לפעולה יש לבצע את השלבים הבאים:
(להכנת דיסקט יש לבצע שלב מספר 3 בלבד).

1. **ביצוע Pre-Format**. תהליך Pre-Format מחלק את הדיסק בצורה פיסית למסלולים וגזרות. ניתן לבצע זאת גם באמצעות תכניות ייעודיות חיצוניות למערכת ההפעלה DOS, כמו Checkit, Disk Manager ועוד.

2. **חלוקת הדיסק לדיסקים לוגיים ויצירת המחיצות** (Partitions) באמצעות תכנית FDISK של מערכת ההפעלה DOS.

3. **ביצוע FORMAT לכל מחיצה** באמצעות מערכת ההפעלה DOS.

שימוש בתכנית כמו Disk-Manager חוסך את ההרצה הנפרדת של שלושת השלבים בנפרד, מכיון שהיא מבצעת את שלושת השלבים ברצף.

5.5 תחזוקת דיסקים קשיחים - הקדם רפואה למכה!

הדיסקים הקשיחים הינם "לב" מערכת המחשב. מחירם גבוה יחסית לשאר יחידות החומרה ובלעדיהם קשה, ולמעשה בלתי אפשרי, לעבוד בצורה ראויה. ההשקעה הרבה הנדרשת להכנסת הנתונים לדיסק הקשיח (מסמכים, נתונים עסקיים וכו') גבוהה עשרות מונים על הסכום שמשקיעים בקניית החומרה למערכת. על כן ראוי שנשים לב לתחזוקה מונעת של הדיסק הקשיח ולאבטחת הנתונים השמורים בו. תחזוקה מונעת וטיפול בדיסק הקשיח, ימנעו אובדן מידע חשוב שלעתים לא ניתן למדוד את ערכו בכסף, ימנעו תקלות מהדיסק הקשיח ויאריכו את חייו.

על מנת לשמור על הנתונים בדיסק הקשיח ועל יעילות השימוש בו מומלץ לבצע את הדברים הבאים:

* גיבוי שוטף

יש לגבות את המידע הנמצא על גבי הדיסק הקשיח באמצעות סרט גיבוי או באמצעות דיסקטים. השימוש בדיסקטים אינו נוח בגלל הטרחה הרבה הכרוכה בכך ולעתים – גם אינה מעשית בגלל מספר הדיסקטים הרב שיש להשתמש בהם עבור דיסק גדול. למשתמשים אשר מהירות הגיבוי ואמינותו חשובים מומלץ לרכוש כונן קלטות לגיבוי (Tape), המאפשר ביצוע של הגיבוי על גבי קלטות מגנטיות במהירות גדולה יותר מגיבוי על גבי דיסקטים ובדרך נוחה יותר.

הגיבוי יכול להיות יומי, שבועי או חודשי והוא נקבע על פי כמות הפעילות וסוג הפעילות המתבצעת בנתונים. מומלץ לשמור אחת לתקופה גיבוי שלם במקום נפרד ואת הגיבוי השוטף לבצע במחזורים של 3 גיבויים (סבא-אבא-בן). כלומר, גיבוי ראשון נעשה על עותק אחד, גיבוי שני על עותק שני, גיבוי שלישי על עותק שלישי, גיבוי רביעי שוב על גבי העותק הראשון וחוזר חלילה. בדרך זו לא נהרוס את עותקי הגיבוי, ואם גילינו תקלה - נוכל לחזור תמיד לגירסה תקינה של הנתונים (גם אם חלקם אבד).

את הגיבוי ניתן לעשות באמצעות העתקה רגילה (COPY, או XCOPY), באמצעות תכנית BACKUP של מערכת ההפעלה DOS, או באמצעות תכניות גיבוי אחרות של תוכנות שירות כמו Norton, PC-Tools, או באמצעות תכניות ייעודיות כמו Fast-Back, למשל.

* הכנה של הדיסק לעבודה

תכניות Pre-Format, FDISK, FORMAT יש להפעיל **רק לאחר גיבוי**. מומלץ להריץ תהליך זה אחת לכמה חודשים, כדי לארגן מחדש את הנתונים על גבי הדיסק. כאשר הקבצים יוחזרו לדיסק בפקודה RESTORE, ההקצאה שלהם תיעשה באופן כזה, שכל קובץ יימצא על הדיסק ברצף של גזרות ומסלולים עוקבים ועל ידי כך לשפר את זמן הטעינה והחיפוש בקבצים. אפשר גם לשנות את ההגדרות המקוריות של הדיסק. ניתן לבצע ארגון מחדש של הדיסק, כלומר הקצאה רציפה של הקבצים, גם כאשר הנתונים רשומים בו. לשם כך יש להשתמש בתכניות שירות כמו SpeedDisk של Norton, Compress של PC-Tools ואחרות.

* הרצת תכנית Chkdsk

זוהי פקודה חיצונית של מערכת ההפעלה DOS. רצוי להפעיל אותה בכל יום, או כל מספר ימים, כדי לבדוק את תקינות ההקצאה של Allocation Units בדיסק. בדיקה באמצעות Chkdsk מאפשרת לזהות תקלות וגם לתקן אותן באופן אוטומטי. אפשר להשתמש בתכניות השירות Disk Doctor, Norton, PC-Tools ואחרות.

* ניהול אחוז תפוסה

המנע ממצב שבו הדיסק מלא בלמעלה מ-90 אחוזים מהקיבולת שלו. במצב זה סיכויי הפגיעה ב-FAT ובמערכת הקבצים גדולים, ונתונים עלולים להיעלם.

* שמירה על נתוני הדיסק העיקריים

רצוי לשמור את גזרת BOOT (Boot Sector) ואת טבלת המחיצות של הדיסק המפרטת את המבנה הלוגי של דיסק פיסי (Partition Table) על גבי דיסקט נפרד. תחת DOS-5 ניתן לשמור פרמטרים אלה באמצעות הפקודה MIRROR. ניתן לבצע שמירה כזאת גם באמצעות תכניות שירות כמו Norton, או PC-Tools. אם קורה נזק לאיזורים אלה על ידי וירוסים למיניהם, הפסקות מתח וכו', אפשר "לאבד" את הקבצים שבדיסק. לכן, אם קיים גיבוי שוטף למידע חשוב זה, ניתן לשחזר אותו ו"להציל" את הקבצים. יש הממליצים גם על שמירה של טבלאות הקצאת הקבצים (FAT).

מכיון שהשינויים בטבלאות אלו הינם תכופים ומתבצעים במהלך העבודה, יש צורך לשמור את הטבלאות באופן שוטף, דבר שאינו מעשי.

* שיפור מהירות העבודה של הדיסק

תוך כדי עבודה שוטפת נוצרות בדיסק מספר תופעות שגורמות להורדת הביצועים שלו. תשומת לב נאותה וטיפול שוטף יכולים לשפר במידה רבה את ביצועי מערכת הדיסקים ובכך להגביר את ביצועי מערכת המחשב כולה. התופעות הן:

קיטוע קבצים (File Fragmentation) – אלגוריתם ההקצאה מקצה איזורים פנויים ולאו דווקא רציפים. אי-רציפות של יחידות ההקצאה (Allocation Units) בקובץ גורמת להגדלת זמן הקריאה של הקובץ בשל התנועות המכניות הרבות שהראש הקורא צריך לבצע כדי להשיג את קטעי הקובץ הפזורים לעתים על פני צלינדרים שונים. המצב העדיף הוא, שכל קובץ ימצא ברצף על גבי הדיסק. ניתן לבדוק רציפות של הקצאת מקום לקבצים באמצעות הפקודה CHKDSK בגירסה 5 ומעלה של DOS.

קיטוע איזורים פנויים (Disk Fragmentation) – מחיקת קבצים יוצרת בדיסק איזורים פנויים, אשר אינם בהכרח רצופים. כתוצאה, הקצאה של יחידות הקצאה לקבצים תיצור בעיה של אי-רציפות. המצב העדיף הוא רצף אחד של כל יחידות ההקצאה בדיסק, כדי שכל הקצאה חדשה תהיה ברצף.

קיטוע ספריות (Directory Fragmentation) – קבצים שונים השייכים לאותה ספריה (Directory) מפוזרים על גבי הדיסק. כתוצאה, על מנת לקרוא קובצי נתונים ותכניות המצויים באותה ספריה על הראש הקורא לבצע תנועות רבות. המצב העדיף הוא, שכל הקבצים השייכים לספריה אחת יהיו ברצף, כדי להקטין את זמן הגישה אליהם.

מצב לא יעיל זה של עבודת הדיסק הקשיח נוצר תוך כדי עבודה. סידור מחדש של הקבצים יכול להיעשות בתהליך BACKUP, FORMAT של הדיסק ו-RESTORE, אך הוא גוזל זמן רב. קיימות תכניות שירות רבות המבצעות פעולות אלו בצורה יעילה ופשוטה להפעלה, כמו Speedisk של Norton, או תכניות השירות של PC-Tools. מומלץ לבצע גיבוי של הנתונים לפני כל פעולה אשר מוחקת, או עלולה לפגוע בנתונים שבדיסק. ארגון מחדש של הדיסק מתקן מצב קיים, אך אינו פותר את הבעיה לתמיד, בגלל ההקצאה של שטחי אחסון וביטולם הנדרשים לכל עדכון של קובץ נתונים או תכנית. לכן יש לחזור על פעולות אלו בצורה שוטפת, בהתאם למידת הפעילות בקובצי הנתונים והתכניות.

* חניית הדיסק - PARK

רוב כונני הדיסק הקיימים בשוק אינם עמידים בפני זעזועים מכניים. על כן, כאשר נטלטל את המחשב, יפגע הראש הקורא ויפגום במשטח המגנטי שמתחתיו. כדי למנוע פגיעה זו בזמן טלטול המחשב ממקום למקום, חייבים "להחנות" את הראש הקורא בסוף המשטח המגנטי. ישנן תכניות PARK שמסופקות יחד עם DOS (ראה תכנית דוגמה ל-PARK בנספח).

5.6 מחיקת קבצים והיחלצות ממחיקה מוטעית

הקצאת מקום פנוי כדי לכתוב בו נתונים נעשית על ידי חיפוש של יחידת הקצאה פנויה החל מהמקום הראשון בטבלת FAT. עבור כל קובץ מוקצים 32 בתים בספריה, או בתת-ספריה, המתארים אותו (שם, תאריך, זמן, מיקום של Allocation Unit ראשון וכו'). מחיקת קובץ אינה מוחקת את הפרטים הרשומים בספריה (Directory), אלא מסמנת בכניסה בטבלה המתייחסת לקובץ שהוא נמחק מהספריה. שים לב שקובץ שנמחק מזוהה באמצעות התו "?" במקום התו הראשון של שמו ועל כן ניתן לשקם אותו באמצעות תכניות שירות. במקביל לפעולת סימון הקובץ בספריה, מסמנת מערכת ההפעלה בטבלת FAT שכל ה-Allocation Units שהוקצו לקובץ פנויים עתה. היא עושה זאת על ידי כתיבת הערך 0 בכל כניסה של FAT בכל אחד משני העותקים שלו.

שיטת סימון זו מאפשרת על כן לשחזר את תוכן הקובץ לאחר מחיקה, כל עוד לא הוקצו יחידות ההקצאה של הקובץ שנמחק לקובץ אחר כלשהו וכל עוד לא נכתב בהן דבר. כלומר, השחזור אפשרי רק אם המשתמש זיהה מיד כי מחק בטעות את הקובץ, או קבצים, ולא המשיך בעבודה. השחזור יכול להיעשות באמצעות תכניות כדוגמת Norton או PC-Tools, או בעזרת הפקודה UNDELETE בגירסה DOS-5 ומעלה. פעולת השחזור כרוכה במספר שלבים:

* **שחזור התו הראשון של שם הקובץ.** כזכור, תו "?" בראש השם מציין קובץ מחוק. כאשר משתמשים בתוכנה PC-Tools-6 ניתן לשמור מידע על כל הקבצים שנמחקו (MIRROR), כדי שאפשר יהיה לשחזר אותם בפעולת UNDELETE. שירות זה ניתן עתה גם על ידי מערכת DOS גירסה 5.

* **מציאת יחידת ההקצאה (Allocation Unit) הראשונה של הקובץ.** מערכת ההפעלה אינה מוחקת פרט זה כאשר היא מציינת שהקובץ נמחק. אם יחידה זו רשומה כתפוסה בטבלת FAT, לא ניתן לשחזר את הקובץ, כי ברור שקובץ אחר נכתב באותו מקום.

* **הקצאת יחידות ההקצאה הפנויות העוקבות ליחידה הראשונה השייכת לקובץ ב-FAT.** מספר יחידות ההקצאה שיוקצו לקובץ מחדש ייעשה על פי גודלו המקורי של הקובץ. גם את גודל הקובץ צריך לעדכן בספריה. ההקצאה של היחידות הפנויות יכולה להיעשות בשתי דרכים:

אוטומטית - באופן פעולה זה מקצים את כל יחידות ההקצאה בצורה אוטומטית בזו אחר זו על פי סדר היחידות העוקבות הפנויות ב-FAT. מניחים כי מחיקת הקובץ יצרה שרשרת של יחידות הקצאה פנויות ולכן הקצאת יחידות עוקבות פנויות על פי סדרן תאפשר, קרוב לודאי, שחזור אמין של הקובץ שנמחק. במקרה שנמחקו מספר קבצים שהיו משולבים אחד בשני (מבחינת מקום פיסי על הדיסק), יש לבצע שחזור ידני.

ידנית (Manual) - באופן פעולה ידני המשתמש בוחר את יחידות ההקצאה השונות עבור הקובץ. הוא נדרש לבחון ולאשר כל יחידת הקצאה שהמערכת בחרה להוסיף לקובץ המשוחזר. גם כאן יחידת ההקצאה התורנית תהא יחידת ההקצאה הפנויה הבאה. נוהגים בדרך זו כאשר השחזור האוטומטי לא הצליח לשקם את הקובץ. הדבר יכול לקרות כאשר ההקצאה של היחידות

אינה רציפה וישנן ב-FAT יחידות הקצאה פנויות נוספות כתוצאה ממחיקת קבצים אחרים.

הערות:

* **אם יחידות ההקצאה שהוקצו לקובץ בעבר כבר הוקצו מחדש, אין כל דרך לשחזר את תוכנו המלא של הקובץ שנמחק. מסיבה זו יש לשמור תמיד גיבוי כדי למנוע אובדן של נתונים ותכניות.** שימוש בתכניות שחזור טוב ויעיל רק אם המחיקה נתגלתה תוך פרק זמן קצר בלא שהמחשב הקים קבצים חדשים במקום שהתפנה על ידי הקובץ שנמחק. גילוי של מחיקה מוטעית לאחר זמן רב לא תאפשר שחזור.

* תוכנות כמו Norton או PC-Tools מספקות תכניות תושבות (Resident) בזיכרון, אשר רושמות את כל הפרטים על הקבצים שנמחקו במערכת, על מנת לאפשר שחזור אמין במקרה של מחיקה מוטעית. בצד היתרון שבשמירת מידע זה, יש לזכור כי תכניות אלו גוזלות זמן מחשב ושהמידע הזה שמיש כל עוד השטח שהתפנה לא הוקצה לקבצים אחרים שהוקמו.

5.7 שיפור הביצועים של מערכת הדיסקים

5.7.1 תכנית זיכרון מטמון לדיסק – Disk Cache

תכנית זיכרון מטמון לדיסק (Disk Cache) מאפשרת שיפור של ביצועי הדיסק בצורה משמעותית (נושא "זיכרון מטמון" נדון בהרחבה בסעיף 2.5.4). זיכרון זה משמש כתווך בין הזיכרון הראשי לבין הדיסק, שהוא התקן איטי ביחס לזיכרון הראשי. רצוי שבזיכרון המטמון יהיו כמה שיותר נתונים שיובאו מראש מהדיסק, כדי שיהיו זמינים למעבד. השיטה ליישום של מטמון הדיסק היא על ידי הקצאה של מקום בזיכרון המחשב, ועדיף בזיכרון ההרחבה שמעל 640KByte, כדי לא לבזבז מקום בזיכרון הראשי.

כאשר מערכת ההפעלה מבקשת לקרוא מהדיסק גזרה מסוימת, היא בודקת תחילה אם התוכן שלה נמצא בזיכרון המטמון (Cache). אם הנתונים נמצאים במטמון, היא תקרא אותם לשטח העבודה ואם הם לא נמצאים בו – היא תקרא אותם מהדיסק. הקריאה תבוצע ישירות לזיכרון הראשי ובמקביל – גם לזיכרון המטמון. מעתה עותק של הגזרה נמצא בזיכרון המטמון. אם מערכת ההפעלה תבקש לקרוא את אותה גזרה שנית מהדיסק, תכנית זיכרון המטמון תמצא את העותק בזיכרון המטמון ותעביר אותו משם לזיכרון הראשי ללא צורך בפניה לדיסק.

כאשר עומד לרשותנו זיכרון מטמון גדול, ניווכח שרוב הפניות לקבצים, ובמיוחד קבצים גדולים, מתבצעות במהירות גבוהה מאוד. כאשר זיכרון המטמון גדול מספיק, לא נדרשות גישות לדיסק כדי לקרוא נתונים ועל כן התכנית תפעל במהירות גדולה. עם מערכת ההפעלה DOS מסופקת יחד תכנית שירות Disk Cache הנקראת SMARTDRV. תכנית זו יש לטעון באמצעות פקודה בקובץ CONFIG.SYS. תכנית זיכרון המטמון של מערכת ההפעלה משתמשת בזיכרון ההרחבה Extended או Expanded (עדיף!) ולא בזיכרון הבסיסי שמתחת ל-640KByte.

ישנן תכניות Disk-Cache המאפשרות שימוש ב-Cache גם לצורכי כתיבה לקובץ. בצורה זו ייכתב מחדש תוכן של גזרה (Sector) על גבי הדיסק רק אם תוכנה הנמצא בזיכרון Cache שונה מהתוכן שיש לכתוב על גבי הדיסק. הפעלת Cache גם לכתיבה משפרת את זמן ריצת התכניות, אך מצב זה לא תמיד רצוי ועלול לפגוע בנתונים שבדיסק. לדוגמה, בהפסקת חשמל, כאשר תכנית היישומים כאילו כתבה לדיסק את הנתונים, אבל הם נרשמו למעשה בזיכרון המטמון. ישנה תמיד אפשרות לבטל את השימוש בזיכרון המטמון.

בחלק מהמחשבים שמותקן בהם מעבד 80386SX, 80386DX או 80486, קיימת ב-ROM BIOS תכנית Disk Cache המאפשרת להקצות חלק מזיכרון ההרחבה שלהם לטובת זיכרון מטמון לדיסק ולהשיג בכך ביצועים טובים יותר.

5.7.2 בקרי דיסק בעלי זיכרון מטמון
Hard Disk Cache Controllers

בקר הדיסק שולט על פעולות קלט ופלט המתבצעות בדיסק. ספקים שונים הוסיפו לו גם את הפונקציה של ניהול זיכרון מטמון, שאינו אלא זיכרון נוסף, אשר משמש כחיץ מהיר בין הכונן הפיסי לבין הזיכרון הראשי של המחשב. ההבדל בין בקר בעל זיכרון מטמון לבין תכנית זיכרון מטמון הוא בכך, שבבקר מטמון נעשית כל הפעולה באמצעות כרטיס חומרה ולא באמצעות תכנית מחשב הגוזלת משאבים ומשפיעה על ביצועי המחשב. כרטיס חומרה של בקר עם זיכרון מטמון זמין רק במחשבי AT ומעלה. זיכרון המטמון הינו בגודל של 256KByte ועד 16MByte.

הבקרים תומכים בדרך כלל גם בכונני הדיסקים וגם בכונני הדיסקטים. בקרים מסוג זה נפוצים יותר בדיסקים גדולים מאוד שפועלים בממשק ESDI ו-SCSI (ראה סעיף 5.1.1). הזיכרון צרוב על גבי ROM בכרטיס הבקר (DISK ROM BIOS). טעינת תכנית ROM לניהול הדיסק מתבצעת בשיטה שהוזכרה לגבי מחשבים המבוססים על מעבד 8088. תכנית ההפעלה שעל הבקר דרושה מכיון שתכנית ROM BIOS של המחשב אינה מכירה סוג זה של בקרים.

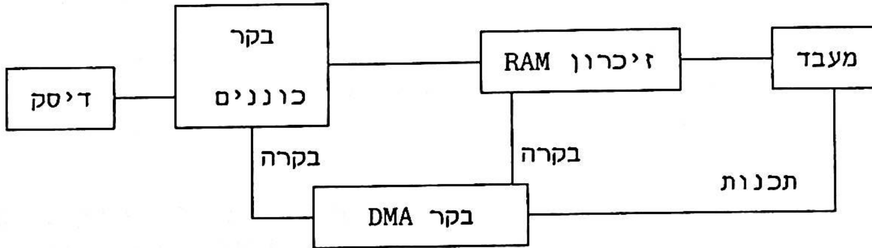
כמה מהבקרים מסוג זה יכולים לזכור את מקומו של הראש הקורא לאחר פעולה אחרונה. על כן, אם ישנן מספר בקשות לקריאה ממסלולים שונים בדיסק, הם מתכננים את הבאת הנתונים בצורה המהירה ביותר, במינימום תנועות הראש הקורא ובכך הם משפרים את ביצועי מערכת הדיסקים.

בדיקת ביצועים של בקרי דיסק עם זיכרון מטמון מעלה שבמרבית המקרים אין ביצועי המחשב עם בקר כזה גבוהים מביצועי המחשב עם תוכנת Disk Cache. הדבר נובע בעיקר מכך, שמערכת ההפעלה DOS מכוונת למשתמש יחיד ומשימה יחידה (Single Task, Single User). אולם, בעבודה במערכות הפעלה התומכות במספר משימות, או תחת **שרת קבצים** (File Server) של רשת תקשורת, יש לבקר כזה יתרון רב על פני תכניות Cache. מחירו של הכרטיס גבוה יותר מההשקעה הדרושה בזיכרון הרחבה על גבי הלוח הראשי, ולכן בשימוש מקובל במערכת הפעלה DOS לא תמיד כדאי לרוכשו.

התרשימים הבאים מתארים מערכות מחשבים הכוללות Disk Cache בתוכנה ובחומרה:

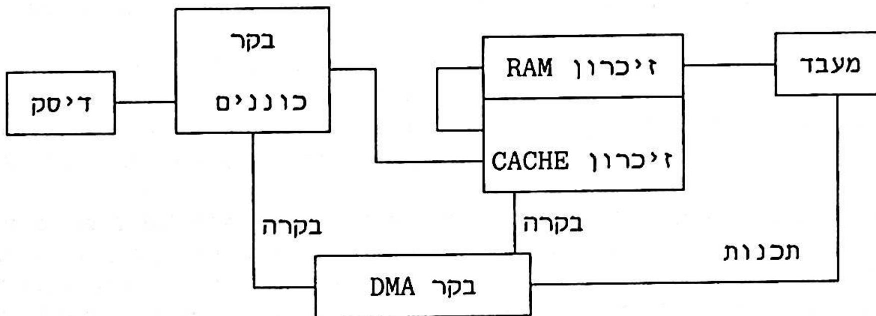
מערכת בסיסית ללא Disk Cache

העברת הנתונים בין בקר הדיסק לבין הזיכרון הראשי נעשית על ידי פעולת DMA. המעבד המרכזי מתכנת את בקר DMA ולאחר מכן הנתונים עוברים ישירות לזיכרון ללא התערבות המעבד.



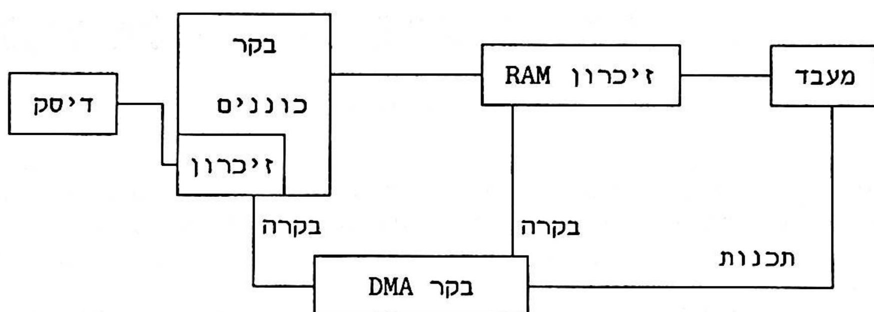
מערכת הכוללת תכנית Disk Cache

ניהול זיכרון המטמון נעשה בתכנית כמו SMARTDRV.SYS. העברת הנתונים בין בקר הדיסק לבין הזיכרון הראשי נעשית על ידי פעולת DMA. העברת נתונים לזיכרון המטמון נעשית על ידי שגרה של ROM BIOS, כאשר זיכרון המטמון נמצא בזיכרון ההרחבה. המעבד המרכזי מתכנת את בקר DMA ולאחר מכן הנתונים עוברים ישירות לזיכרון ללא התערבותו. כאשר פונים לדיסק, תכנית זיכרון המטמון בודקת תחילה אם הנתונים נמצאים בזיכרון המטמון. אם הם נמצאים בו, היא מביאה אותם מזיכרון המטמון, אחרת, הם מובאים מהדיסק.



מערכת הכוללת בקר בעל זיכרון מטמון (Disk Cache)

כאשר בקר הדיסקים כולל זיכרון המטמון, העברת הנתונים בינו לבין הזיכרון הראשי נעשית באמצעות פעולת DMA. המעבד המרכזי מתכנת את בקר DMA ולאחר מכן הנתונים עוברים ישירות לזיכרון ללא התערבותו. כאשר הנתונים נמצאים בזיכרון המטמון שבבקר הם מועברים ישירות לזיכרון המחשב, ואם הם אינם בזיכרון המטמון – הם מובאים מהדיסק.



5.8 דיסק בפועל – Virtual Disk

תפישת העבודה בשיטת "דיסק בפועל" מאפשרת להגדיר חלק מזיכרון המחשב כדיסק לוגי. עדיף להשתמש לצורך זה בזיכרון הרחבה, כדי לא לגזול מקום בזיכרון הראשי. לשימוש בזיכרון RAM כדיסק יש חסרונות ויתרונות של זיכרון, כמו מהירות העברה ונדיפות!

בדיסק בפועל ניתן לשמור נתונים כאילו היה דיסק אמיתי ובאותו מבנה ואופן שהמערכת פועלת על דיסק פיסי. היתרון שלו הוא בכך שכל פעולות הקלט/פלט נעשות במהירות RAM ועל כן תכנית שבה יש פעולות קלט/ פלט רבות תפעל במהירות רבה יותר. החיסרון של השימוש בדיסק בפועל הוא בכך שיש צורך להעתיק אליו את כל הנתונים לפני תחילת העבודה ולהחזיר אותם לדיסק הפיסי בגמר העבודה.

שימוש בדיסק בפועל לאחסון זמני של נתונים ופלט מתכניות מחייב זהירות רבה, מכיון שהשמירה נעשית ב-RAM של המחשב וכל תקלה בתוכנה, כיבוי המחשב, או נפילת מתח עלולים למחוק את תוכן הזיכרון **ולגרום לאובדן הנתונים של "הדיסק" ושל שעות עבודה רבות.**

מערכת ההפעלה מתייחסת לדיסק בפועל כדיסק פיסי רגיל. היא מספקת תכנית להפעלתו באמצעות Driver הנקרא VDISK.SYS. פקודה זו יש להגדיר בקובץ CONFIG.SYS. בגירסה 4 ומעלה של DOS נשתמש בתכנית RAMDRIVE.SYS.

את מספר הכניסות בספריית השורש של דיסק בפועל ניתן לכוון. מערכת ההפעלה מקצה עותק אחד בלבד של FAT לדיסק זה כדי לחסוך במקום, ובשל האמינות הגדולה יותר של הזיכרון הראשי במחשב, לעומת שיטת הרישום על גבי דיסקים/דיסקטים.

5.9 פקודות DOS לטיפול במערכת הדיסקים/דיסקטים

5.9.1 תכנית FDISK

בגמר תהליך Pre-Format לדיסק הקשיח יש ליצור טבלת מחיצות. כדי לעשות זאת, יש לבצע BOOT מכונן A עם דיסקט מערכת הפעלה המכיל את הקובץ FDISK.EXE.

לאחר שמקישים את הפקודה FDISK מקבלים תפריט שבו 4 אפשרויות:

1. Create DOS Partition
2. Change Active Partition
3. Delete DOS Partition
4. Display Partition Data

[5. Select Next Fixed Disk]

אפשרות 5 תופיע רק במערכת עם שני דיסקים קשיחים.

הסברים לאפשרויות הבחירה:

1 - יצירת מחיצת DOS ראשית (Primary), או מורחבת (Extended). בכונן C יש ליצור תחילה מחיצת DOS ראשית ורק לאחר מכן ניתן ליצור מחיצת DOS מורחבת. בכונן D ניתן ליצור את מחיצת DOS המורחבת גם ללא מחיצה ראשית.

2 - בכונן C יש ליצור את אחת המחיצות כמחיצה שניתן לבצע דרכה BOOT. אם לא נעשה זאת, נקבל הודעת שגיאה (Invalid Partition Table) בתהליך BOOT.

3 - מחיקת מחיצות שנוצרו. זוהי פעולה הרסנית, אשר מוחקת את תיאור פרטי המחיצה.

4 - הצגת מבנה המחיצות המוגדרות בדיסק.

5 - בחירה של דיסק D לצורך חלוקה למחיצות. אפשרות בחירה זו לא תופיע אם קיים רק דיסק יחיד במערכת.

תהליך העבודה:

המשתמש צריך לקבוע אם רצונו במחיצה אחת או יותר וכיצד הוא רוצה לחלק את הדיסק למחיצות. אם החליט על מחיצה אחת, עליו ליצור מחיצת DOS ראשונית (Primary) על ידי בחירה באפשרות 1 שבתפריט הראשי. בתפריט משני של יצירת מחיצות עליו לבחור באפשרות 1 ליצירת מחיצה ראשית (Primary) ולהקצות למחיצה זו את כל המקום בדיסק. התכנית FDISK תהפוך את המחיצה היחידה בדיסק למחיצה שניתן לבצע דרכה את תהליך BOOT. עתה יוצאים מתכנית FDISK וממשיכים בהכנת הדיסק לעבודה.

אם החליט לחלק את הדיסק למספר מחיצות, עליו לבצע מספר פעולות. לצורך הדוגמה נניח שלפנינו דיסק של 20MByte עם 613 מסלולים, שיש לחלקו ל-3 מחיצות שוות, כך שנקצה כ-200 מסלולים לכל מחיצה. המשתמש צריך:

* ליצור תחילה מחיצת DOS ראשית על פי תהליך שתואר בשלב קודם ולהקצות לה 200 מסלולים ולא את כל שטח הדיסק. לחזור לתפריט ראשי על ידי הקשת Esc.

* לבחור שוב ביצירת מחיצה (אפשרות 1) ובתפריט המשני של יצירת מחיצות לבחור יצירת מחיצת DOS מורחבת (Extended) (אפשרות 2 בתפריט). יש להקצות למחיצה זו את כל המקום שנותר בדיסק (413 מסלולים). יש לחזור לתפריט הראשי על ידי הקשת Esc.

* לבחור שוב ביצירת מחיצה (אפשרות 1 בתפריט ראשי). בתפריט המשני יש לבחור באפשרות 3 ליצירת דיסקים לוגיים על מנת להגדיר את הדיסקים הלוגיים בתוך המחיצה המורחבת. כאן יש להגדיר לכל דיסק את הגודל המבוקש (במקרה זה 200 מסלולים לראשון ו-213 מסלולים לשני). חזרה לתפריט ראשי על ידי הקשת Esc.

* כל שנותר בשלב זה הוא להגדיר את מחיצת DOS הראשית לביצוע BOOT (Active). יש לבחור באפשרות 2 בתפריט הראשי ולסמן את מחיצה מספר 1 הראשית כ-Active. עתה הדיסק מוכן להמשך העבודה (פעולת FORMAT).

5.9.2 FORMAT - הכנת דיסק/דיסקט לעבודה במערכת ההפעלה

הפקודה - FORMAT
הכנת דיסק/דיסקט לעבודה תחת מערכת ההפעלה DOS.

מקום הפקודה - מערכת ההפעלה (פקודה חיצונית).

מבנה הפקודה -

FORMAT D:[/1][/4][/8][/n:Sectors][/t:Tracks][/v:Label][/s][/b]

או: FORMAT D:[/1][/4][/8][/v:Label][/s][/b][/f:Size]

כאשר:

/1 - בצע FORMAT לצד אחד בלבד (דיסקט עם משטח מגנטי מצד אחד בלבד). אפשרי בדיסקטים "5.25 בקיבולת של עד 360KByte.

/4 - ביצוע FORMAT לדיסקט "5.25 בקיבולת של 360KByte על ידי כונן בעל צפיפות גבוהה של 1.2MByte.

/8 - ביצוע FORMAT של דיסקטים בקיבולת של עד 360KByte במבנה של 8 גזרות במסלול במקום 9 גזרות (לצורכי תאימות עם כוננים ישנים ומערכות הפעלה מגירסה ישנה).

/t:Tracks - הגדרה של מספר המסלולים במשטח המגנטי. ההגדרה אפשרית בדיסקטים "3.5 בלבד.

/n:Sectors - הגדרה של מספר גזרות בכל מסלול. ההגדרה אפשרית בדיסקטים "3.5 בלבד.

לדיסקט 3.5 בקיבולת 720K כתוב: /t:80/n:9
לקיבולת 1.44MB כתוב: /t:80/n:18

הערה: נשתמש בשיטה זו בגירסאות DOS 3 ומטה ולכוננים מיוחדים.

/v:Label - הגדרה של תווית Label (שם כלשהו עד 11 תווים) לדיסק, או לדיסקט. אם לא תציין כאן את השם, אך תציין את הפרמטר v/, יחכה המחשב לנתון זה בסיום פעולת FORMAT.

/b - בצע FORMAT ללא גרעין מערכת הפעלה, אך השאר מקום פנוי עבור מערכת הפעלה שתוכנס בעתיד. בגירסאות קודמות לגירסה 4 של DOS, לא ניתן היה להוסיף לדיסקט את מערכת ההפעלה אם לא ביקשנו זאת מראש בעת FORMAT.

/s - בגמר תהליך FORMAT הכנס לדיסקט את גרעין מערכת ההפעלה. יש לזכור, כי גרעין מערכת ההפעלה תופס מקום על חשבון קבצים ולכן נשתמש באפשרות זו רק כאשר יש צורך לאפשר BOOT דרך הדיסק/דיסקט שמכינים לעבודה.

/f:Size - בצע FORMAT לדיסקט בקיבולת Size. הקיבולת הסטנדרטית של הדיסקט קובעת את הפרמטרים (צדדים, מסלולים, גזרות) ולכן אין צורך להגדירם. אין אפשרות להגדיר פרמטרים n ו-t של דיסקט יחד עם אפשרות זאת.

Size -	עבור "5.25:	160	180	320	360	1,200
(KBytes)	עבור "3.5:	360	720	1,440		

/q - אתחול מהיר. הפקודה משתמשת במבנה הקיים של הדיסקט, כשהתנאי הוא שהדיסקט עבר אתחול לפחות פעם אחת.

/u - אתחול ללא תנאי. יתבצע אתחול לדיסקט מבלי לבדוק הימצאות של נתונים. הפעולה בלתי הפיכה.

דוגמאות:

הפקודה FORMAT A:/4/S

הכנת דיסקט של 360KByte בכונן של 1.2MByte, אשר יכלול את גרעין מערכת הפעלה.

הפקודה FORMAT C:/B/S/V:example

הכנת הדיסק הקשיח C לעבודה. יש להשאיר מקום להכנסה עתידית של מערכת ההפעלה. שם הדיסק הוא EXAMPLE.

הפקודה FORMAT B:

אם הפקודה FORMAT מופנית לכונן דיסקט שקיבולתו 1.44MByte, היא לא תשאיר מקום למערכת ההפעלה, אך תכין את כל איזורי מערכת ההפעלה (Root Directory, FAT). היא תערוך את הדיסקט במבנה בעל 80 מסלולים, 2

צדדים, 18 גזרות בכל מסלול. אם נכניס לכונן זה דיסקט של 720KByte, נקבל הודעות שגיאה במהלך ביצוע פעולת האתחול. הפקודה הנכונה לאתחול דיסקט של 720K בכונן של 1.44MB צריכה להיות FORMAT b:/f:720.

הערות:
הפקודה FORMAT עד לגירסה 5 של DOS המתבצעת בדיסקט היא **בלתי הפיכה.** פעולת FORMAT מוחקת את כל הנתונים בדיסקט ולא ניתן לשחזר אותם. בדיסק קשיח אובדים כל הקשרים לנתונים, אך אין כתיבה מחדש על גבי הדיסק באיזורי הנתונים. גם כך פעולת השחזור כמעט בלתי אפשרית. בכל מקרה, יש להיזהר בביצוע הפקודה ולבדוק על איזה כונן הפקודה עומדת להתבצע והאם הביצוע תואם את בקשתנו. מערכת ההפעלה תבקש תמיד אישור להמשך הפעולה, קרא אותה בעיון ובדוק אם לא טעית. מגירסה 5 ומעלה של DOS הפקודה FORMAT שומרת את המידע הנחוץ לשחזור הנתונים בדיסק או בדיסקט.

5.9.3 UNDELETE - היחלצות ממחיקה מוטעית של קבצים

הפקודה - UNDELETE
היחלצות ממחיקה מוטעית של קבצים. הפקודה מאפשרת לשחזר קבצים שמחקנו ממערכת הקבצים, בתנאי שלא נרשמו נתונים חדשים באיזור שבו אוחסנו קודם קבצים אלה.

מקום הפקודה - מערכת ההפעלה (פקודה חיצונית).

מבנה הפקודה -

```
UNDELETE [Drive] [Path][File-Names] /Options
```

כאשר Options מתוך האפשרויות הבאות:

/ALL - שחזר את כל הקבצים מבלי לשאול את המשתמש לאישור ולהשלמת האות הראשונה בקובץ. במידה וחסרה האות הראשונה בקובץ (?) האות הראשונה תושלם לסימן # (או & אם כבר קיים קובץ בשם זה).

/DOS - שחזור של הקבצים הנמצאים בספריית MS-DOS בלבד. לא יתבצע חיפוש בספריות אחרות.

/DT - ישוחזרו רק הקבצים הרשומים בקובץ המעקב אחר מחיקות קבצים (File Deletion Tracking). קובץ המעקב נוצר מהפעלת תכנית מעקב MIRROR (ראה הפעלת תכנית מעקב בסעיף 5.9.5).

/LIST - הצג את שמות הקבצים המסומנים כקבצים שנמחקו מהמערכת. יוצגו שמות הקבצים והסיכוי לשחזור שלהם.

הערות:
- אופציות ALL ו-LIST לא ניתן להפעיל ביחד.
- אופציות DT ו-DOS לא ניתן להפעיל ביחד.
- כל שאר הצירופים אפשריים.

דוגמאות:

<u>הפקודה</u> UNDELETE C:\IDO*.*/LIST/DOS

תציג את כל הקבצים הרשומים כמחוקים בספריית C:\IDO אך לא תשחזר אותם. בצד כל קובץ תציג התכנית אם ניתן לשחזר אותו.

```
Directory: C:\IDO
File Specifications: *.*

    Deletion-tracking file not found.
       MS-DOS directory contains   13 deleted files.
    Of those,    7 files may be recovered.

 Using the MS-DOS directory.

   ** ?P09C78F $$$    57344  7-07-91  9:56a  ...A
      ?VALUE   DAT      606  9-10-90  9:17a  ....
      ?P025DAE $$$    65536  7-06-91  3:25a  ...A
      ?RTABLE  BAK     4985  7-06-91  2:47a  ...A
   ** ?SCOP    DAT     9600  1-11-91  8:07p  ....
   ** ????             3166  5-24-91 10:29p  ....
   ** ?DO      SCN     4000  5-06-91  6:12p  ....
      ?P01AFA5 $$$    12288  7-06-91  1:43a  ...A
      ?RTABLE  BAK     4946  7-06-91  1:41a  ...A
   ** ?VDEM    EXE    46160  5-04-91 10:25p  ....
      ?VDEM    PAS     2189  5-04-91 10:24p  ....
   ** ?RLEDS   BAK    10792  7-06-91  1:10a  ...A
      ?RTABLE  BAK     4950  7-06-91  1:23a  ...A

   "**" indicates the first Allocation Unit of the file
       is unavailable and cannot be recovered
       with the UNDELETE command.
```

<u>הפקודה</u> UNDELETE C:\IDO*.*/ALL/DOS

תשחזר באופן אוטומטי את כל הקבצים שניתן לשחזר מבלי לבקש אישור, או אות ראשונה מהמשתמש.

```
Directory: C:\IDO
File Specifications: *.*

    Deletion-tracking file not found.
     MS-DOS directory contains   12 deleted files.

    Of those,    6 files may be recovered.

 Using the MS-DOS directory.
```

```
   ** ?P09C78F $$$     57344  7-07-91  9:56a  ...A
Starting Allocation Unit is unavailable. This file cannot be
recovered with the UNDELETE command.  Press any key to
continue.

      ?VALUE   DAT        606  9-10-90  9:17a  ....

File successfully undeleted.

      ?P025DAE $$$      65536  7-06-91  3:25a  ...A

File successfully undeleted.

      ?RTABLE  BAK       4985  7-06-91  2:47a  ...A

File successfully undeleted.

   ** ?SCOP    DAT       9600  1-11-91  8:07p  ....
Starting Allocation Unit is unavailable. This file cannot be
recovered with the UNDELETE command.  Press any key to
continue.

   ** ????              3166  5-24-91 10:29p  ....
Starting Allocation Unit is unavailable. This file cannot be
recovered with the UNDELETE command.  Press any key to
continue.

   ** ?DO      SCN       4000  5-06-91  6:12p  ....
Starting Allocation Unit is unavailable. This file cannot be
recovered with the UNDELETE command.  Press any key to
continue.

      ?P01AFA5 $$$      12288  7-06-91  1:43a  ...A

File successfully undeleted.

   ** ?VDEM    EXE      46160  5-04-91 10:25p  ....
Starting Allocation Unit is unavailable. This file cannot be
recovered with the UNDELETE command.  Press any key to
continue.

      ?VDEM    PAS       2189  5-04-91 10:24p  ....

File successfully undeleted.

   ** ?RLEDS   BAK      10792  7-06-91  1:10a  ...A
Starting Allocation Unit is unavailable. This file cannot be
recovered with the UNDELETE command.  Press any key to
continue.
```

```
?RTABLE  BAK      4950  7-06-91  1:23a  ...A
File successfully undeleted.
```

הפקודה UNDELETE *.*/DOS

תשחזר את כל הקבצים מהספרים C:\SEMINAR (על פי הסימן המנחה ניתן לראות כי הפקודה בוצעה בספריה זאת). עבור כל קובץ המשתמש נדרש להשלים את האות הראשונה של שם הקובץ במקום סימן השאלה (?).

```
C:\SEMINAR>UNDELETE *.*
Directory: C:\SEMINAR
File Specifications: *.*

    Deletion-tracking file not found.

    MS-DOS directory contains    2 deleted files.
    Of those,    2 files may be recovered.

Using the MS-DOS directory.

?1        BAK    65437  7-04-91 11:14p  ...A  Undelete (Y/N)?y
      Please type the first character for ?1      .BAK: s

File successfully undeleted.

?11       BAK    15881  7-04-91 11:16p  ...A  Undelete (Y/N)?y
      Please type the first character for ?11     .BAK: s

File successfully undeleted.
```

5.9.4 UNFORMAT - היחלצות מאתחול מוטעה של דיסק/דיסקט

הפקודה UNFORMAT - שחזור תוכן דיסק/דיסקט אשר עבר פעולת אתחול באמצעות הפקודה FORMAT.

מקום הפקודה - מערכת ההפעלה (פקודה חיצונית).

מבנה הפקודה - UNFORMAT Drive: /Options

הפרמטר Options נלקח מתוך האפשרויות הבאות:

/J - מתבצע אימות של הקובץ המאפשר שחזור הדיסק/דיסקט MIRROR.FIL לעומת הנתונים הרשומים על גבי הדיסק/דיסקט. אם הנתונים מתאימים השחזור אפשרי בצורה מלאה.

/L - הצגה של הקבצים ותת הספריות שניתן לשחזר מהדיסק/דיסקט. מתבצעת בדיקה מלאה על כל הדיסק/דיסקט על מנת למצוא את כל הנתונים האפשריים. כאשר יתבצע שימוש גם באופציה PARTN תוצג גם טבלת המחיצות המתאימה לדיסק.

/P - כל ההודעות למיניהן של התכנית יישלחו גם למדפסת המקבילית המחוברת לערוץ המקבילי הראשון (LPT1) או להתקן הדפסה אחר במידה ובוצע ניתוב מחדש (Redirection).

/PARTN - שחזור של טבלת המחיצות של הדיסק הקשיח. הדבר יעיל כאשר משתמשים בתכנית FDISK ומשנים את מבנה המחיצות בדיסק. פעולה זו מאפשרת לשחזר את המחיצות ואת מבנה הדיסק.

/TEST - מתבצעת רק בדיקה של אפשרות השחזור ולא שחזור אמיתי. הנתונים של השחזור לא נכתבים לדיסק. הדבר יעיל כאשר מעוניינים לוודא תקינות השחזור לפני השחזור המעשי.

/U - השחזור של הדיסק/דיסקט יתבצע מבלי להשתמש בנתוני קובץ השחזור MIRROR.FIL. יעיל כששאר הקובץ לשחזור ניזוק ומעוניינים לבדוק אפשרות שחזור של הדיסק. אפשרות זאת לא תמיד יעילה!

דוגמה: UNFORMAT B:

פקודה זו תשחזר את הדיסקט מכונן B ותשתמש במידע שבקובץ MIRROR.FIL על מנת לשחזר את הנתונים בדיסקט.

```
Insert disk to rebuild in drive B:
and press ENTER when ready.


Restores the system area of your disk by using the image file
created by the MIRROR command.

    WARNING !!          WARNING !!

This command should be used only to recover from the
inadvertent Use of the FORMAT command or the RECOVER command.
Any other use of the UNFORMAT command may cause you
to lose data!  Files modified since the MIRROR image
file was created may be lost.

Searching disk for MIRROR image.

The last time the MIRROR or FORMAT command was used was
at 10:58 on 07-15-91.

The MIRROR image file has been validated.

Are you sure you want to update the system area of your
```

```
drive B (Y/N)? y

The system area of drive B has been rebuilt.

You may need to restart the system.
```

הערות:

* אם פעולת האיתחול בוצעה ב-DOS גירסה 5 ומעלה, המידע לשחזור נשמר על גבי הדיסק/דיסקט ואם לא נכתב בו דבר, השחזור יתבצע בצורה מלאה! אם האיתחול בוצע בגירסה נמוכה יותר של DOS או על ידי שימוש באופציה U/ לא ניתן לשחזר את תוכן הדיסקט ואילו בדיסק ניתן יהיה לשחזר את מרבית הנתונים.

* האופציה J ניתנת לשימוש רק כאשר היא האופציה היחידה בפקודה.

* האופציה PARTN/ ניתנת לשימוש רק עם האופציה L/.

5.9.5 MIRROR - שמירה של נתוני תפעול חשובים של הדיסק

הפקודה - MIRROR
שמירה של נתוני הדיסק: גיזרת BOOT, עותקי FAT וטבלת מחיצות, על מנת שניתן יהיה לשחזר את הדיסק במקרה של נזק מוירוס, או שימוש מוטעה בפקודות FDISK ו-FORMAT.

מקום הפקודה - מערכת ההפעלה (פקודה חיצונית).

מבנה הפקודה - MIRROR [Drive:]/Options

כאשר Options מתוך האפשרויות הבאות:

1/ - שמירה של המידע החדש שנאגר על גבי הדיסק בלבד ולא לשמור שוב את המידע הקודם.

PARTN/ - שמירה של טבלת המחיצות של הדיסק הקשיח על גבי דיסקט. נחוץ על מנת לגבות את מבנה הדיסק על גבי מדיה מגנטית חיצונית.

TDrive/ - הפעלת תכנית למעקב אחר מחיקת קבצים עבור הדיסק Drive. מרגע זה עבור כל מחיקת קובץ יירשמו כל נתוני הקובץ שנמחק והשחזור יוכל להתבצע בצורה מלאה מבלי שהמשתמש יידרש לזכור את האות הראשונה בקובץ.

Entries-Number - מספר הכניסות המירבי למעקב אחר מחיקת קבצים. בתוך הקובץ יירשמו רק Entries-Number הקבצים האחרונים שנמחקו.

U/ - ביטול והוצאה (UNLOAD) של התכנית למעקב אחר מחיקת קבצים מהזיכרון. יש לזכור כי התכנית צורכת משאבי מחשב (זיכרון וזמן) ולכן לא תמיד נרצה לטעון אותה.

דוגמאות:

<u>הפקודה</u> MIRROR /TC-20

הפקודה תשמור את נתוני הדיסק C בקובץ MIRROR.FIL ותפעיל את תכנית המעקב אחר מחיקת קבצים עבור כונן C. בתכנית המעקב יוגדרו 20 כניסות למעקב.

```
Creates an image of the system area.

Drive C being processed.

The MIRROR process was successful.

Deletion-tracking software being installed.

The following drives are supported:
Drive C -  20 files saved.
    Installation complete.
```

<u>הפקודה</u> MIRROR /PARTN

```
Disk Partition Table saver.

The partition information from your hard drive(s) has been
read.

Next, the file PARTNSAV.FIL will be written to a floppy disk.
Please insert a formatted diskette and type the name of the
diskette drive.
What drive? A

Successful.
```

הפקודה תשמור את טבלת המחיצות של הדיסק על גבי קובץ בדיסקט בכונן A תחת השם PARTNSAV.FIL.

<u>הפקודה</u> MIRROR /U

תסיר את תכנית המעקב מזיכרון המחשב.

```
Deletion-tracking software removed from memory.
```

הערה:
האופציות U/ ו-PARTN/ ניתנות לשימוש רק כאשר הן האופציות היחידות המוזכרות בפקודה.

5.9.6 הגדרת כונן שלא מוכר על ידי ROM BIOS

הפקודה - DRIVER.SYS
חיבור של התקני דיסקים/דיסקטים בעלי תכונות אשר אינן נתמכות על ידי תכנית ROM BIOS.

מקום הפקודה - יש להכניס את הפקודה לקובץ CONFIG.SYS.

מבנה הפקודה -

DEVICE=DRIVER.SYS [/D:n][/c][/F:f][/H:h][/S:s][/T:t]

בגירסה 4 ניתן לרשום:

DRIVPARM=[/D:n][/c][/F:f][/H:h][/S:s][/T:t]

כאשר:

/D:d - איזה כונן פיסי יש להגדיר (0 עבור A, 1 עבור B וכו').

/c - מציין שהכונן יכול להבחין במצב פתיחת דלת של הכונן. יש להגדיר אפשרות זו, רק אם הכונן מזהה את הפעולה.

/F:f - f מציין את סוג הכונן
0 - כונן של 360KByte
1 - כונן של 1.2MByte
2 - כונן של 720KByte
7 - כונן של 1.44MByte

אם מציינים את סוג הכונן אין צורך להגדיר את הפרמטרים הבאים. ברירת המחדל של FORMAT היא 2 (כונן 720KByte).

/H:h - h מציין את מספר הראשים הקוראים שישנם בכונן.

/S:s - s מציין כמה גזרות (Sectors) בכל מסלול.

/T:t - t מציין בכמה מסלולים במשטח תומך הכונן.

דוגמה להגדרת כונן "3.5 720KB במחשב XT:

<u>הפקודה</u> DEVICE=DRIVER.SYS /D:1/F:2

נניח שבמחשב XT מותקן כונן B של 720KByte, אף על פי שכונן B ב-XT מוגדר כסטנדרט ל-360K. הפקודה תיצור כונן לוגי נוסף במערכת (אם קיימים A, B, C היא תיצור גם כונן בשם D) שמופנה פיסית לכונן B. כונן :D החדש יהיה בגודל של 720K. תכונות כונן B תשארנה כפי שהן מוגדרות במערכת ההפעלה (360KByte במחשב עם מעבד 8088). על מנת להשתמש בכונן כפי שהגדרנו, יש תמיד לגשת לכונן שנוצר. עכשיו, הפקודה :Format B תערוך 360K על אותו דיסקט והפקודה :Format D תערוך 720K על הדיסקט.

בגירסה 4 נוכל לכתוב: DRIVPARM=/D:1/F:2

גם כאן הגדרנו את הכונן הפיסי שקשור למחבר B של בקר הכוננים ככונן של 720KByte. אך כאן, שלא כבגירסה 3.X, לא נוצר כונן לוגי חדש והפרמטרים החדשים הם במקום הפרמטרים הקודמים של כונן B, כך שהגישה נוחה יותר. הפניה לכונן נשארת לכונן B עם הפרמטרים החדשים של 720K.

אם ברירת המחדל של DOS לשמות כוננים A עד E אינה מספיקה וצריך ליצור שם לוגי נוסף, יש להשתמש בפקודה LASTDRIVE לצורך הגדרת השם הנוסף.

5.9.7 LASTDRIVE - שינוי מספר כוננים לוגיים אפשריים

הפקודה - LASTDRIVE
הרחבת מספר שמות הכוננים במערכת. הפקודה משנה את ברירת המחדל של מערכת ההפעלה שמאפשרת להכיר ב-5 כוננים A-E.

מקום הפקודה - יש להכניס את הפקודה לקובץ CONFIG.SYS

מבנה הפקודה - LASTDRIVE = C

כאשר:

c - אות בין A ל-Z המסמנת את אות הכונן המירבית שניתן להגדיר במערכת המחשב.

דוגמה: LASTDRIVE = F

פקודה זו תאפשר להגדיר במערכת כוננים מ-A עד F, כלומר עד 6 כוננים פיסיים ולוגיים בלבד.

5.9.8 CHKDSK - בדיקת תקינות הדיסק/דיסקט

הפקודה - CHKDSK
בדיקת בטבלת FAT אם יש טעויות בהקצאת יחידות ההקצאה (Allocation Units) לקבצים, כמו למשל Allocation Unit שהוקצה בטעות פעמיים או יותר. במקרה כזה יוצגו שמות הקבצים שבהם יש שגיאה מסוג זה. התכנית מראה גם את מצב הדיסק: מספר קבצים, שטח פנוי, שטח מוקצה, גודל יחידת ההקצאה בדיסק, גודל זיכרון RAM כולל וגודל זיכרון RAM פנוי.

מקום הפקודה - יש להקיש את הפקודה כשמקבלים את הסימן המנחה (Prompt) של מערכת ההפעלה. ניתן להכניס את הפקודה לקובץ AUTOEXEC.BAT.

מבנה הפקודה - CHKDSK [drive:][file-names][/F][/V]

כאשר:

:drive - שם הכונן שיש לבדוק.

file-names - שמות הקבצים שעבורם יש לבדוק קיטוע (fragmentation).

/F - הוראה לתקן שגיאות הקצאה על הדיסק. הפעולה היא הרסנית לעתים, כי היא מלווה בשינויים ב-FAT. לכן יש להיזהר כאשר משתמשים באופציה זו. בכל מקרה התכנית תשאל לאישור פעולת השינוי כך שניתן להתחרט.

/V - להציג כל קובץ נבדק במהלך הפעולה של התכנית.

דוגמה: הפקודה CHKDSK C:/F/V תבדוק את כונן C, תציג תוך כדי בדיקת שמות כל הקבצים והספריות הנבדקים ותאפשר לתקן את התקלות אם ישנן, לאחר אישור של המשתמש כמובן.

הערות:

* פעולת התכנית כרוכה בשינויים של טבלת ההקצאה FAT ולכן יש להפעילה בזהירות.

* מומלץ שלא להגיע למצב שבו הדיסק הקשיח מלא ביותר מ-90 אחוזים מקיבולתו. במצב כזה פעולת ההקצאה נמשכת זמן רב ויש סיכון של פגיעה בטבלת ההקצאה. עבודה במשך זמן רב על גבי 10 האחוזים הפנויים האחרונים בדיסק עלולה לגרום לשגיאות רבות ב-FAT.

5.10 פקודות DOS לשיפור מהירות גישה לנתונים

מכיון שהזיכרון הראשי של המחשב (RAM) נמחק בכיבוי המתח, שומרת מערכת ההפעלה את כל המידע החיוני למציאת הנתונים וההתמצאות בדיסק על גבי הדיסק (FAT, Directory). כל בקשה ממערכת ההפעלה להפעיל על מערכת הקבצים בדיסק, מחייבת את מערכת ההפעלה לגשת לדיסק/דיסקט על מנת לחפש בו את הנתונים. הקטנה של מספר הגישות לקריאה מהדיסק על מנת לקרוא מידע הנדרש למערכת ההפעלה, תביא לחיסכון בזמן והביצועים של מערכת המחשב יעלו.

על מנת לשפר את ביצועי המערכת כוללת מערכת ההפעלה DOS מספר פקודות המאפשרות לה לחסוך בגישות לדיסק. כאשר מערכת ההפעלה מתחילה לפעול בסוף תהליך BOOT, מקבלת סיבית הפעולה של המערכת את ערכי ברירת מחדל שקבע יצרן מערכת ההפעלה. ערכים אלה אינם מתאימים לכל סביבות העבודה ועל כן צריך להפעיל מספר פקודות כדי לשנות את סביבת העבודה של מערכת ההפעלה לשיפור ביצועי מערכת המחשב.

5.10.1 הגדרת BUFFERS

הפקודה - BUFFERS

הפקודה מקצה למערכת ההפעלה מקום שבו תוכל לשמור מידע הנחוץ לה לפעולה. מידע זה קיים על גבי המדיה המגנטית במערכת: ספריות, FAT וכו'. הוא משתנה תוך כדי פעילות של מערכת המחשב ועל כן הימצאותו ב-RAM חוסכת גישות לדיסק.

מקום הפקודה - יש לרשום את הפקודה בקובץ CONFIG.SYS.

מבנה הפקודה - BUFFERS=size [,sectors] [/x]

כאשר:

size - מספר הבלוקים בזיכרון של 512 בתים שיש להגדיר לשימוש מערכת ההפעלה DOS. ערכי SIZE הם בין 1 ל-99, אם נקצה את הזיכרון עבור Buffers על גבי הזיכרון הראשי (כלומר נקצה עד 50K). אם נקצה את הזיכרון על גבי זיכרון ההרחבה (/x עבור Expanded, מגירסה 4 ומעלה) הפרמטר יקבל ערכים בין 1 ל-10,000, כלומר עד 5M זיכרון.

sectors - מספר הגזרות שניתן לכתוב או לקרוא בפעולת קלט/פלט אחת בגישה לדיסק. ברירת מחדל של המערכת – 1, הערך שניתן להגדיר הינו בין 1 ל-8.

דוגמה: BUFFERS=30

הקצאה של 30 גזרות לשמירת נתונים לשימוש מערכת ההפעלה.

הערות:

אין להבין מכך, כי תמיד יש לציין את הערכים הגבוהים יותר. אם מקום ההקצאה הוא בזיכרון הראשי, הרי כל BUFFER יגזול ממנו כ-528 בתים. כמו כן, לחיפוש בתוך ה-BUFFERS יש מחיר של זמן ולכן מעל גודל מסוים לא יהיה חיסכון בזמן, אלא להיפך. יש לבצע מספר נסיונות עם ערכים שונים של BUFFERS ולראות כיצד המערכת מגיבה. גודל מומלץ 30 עד 40.

5.10.2 FASTOPEN - שמירת כתובות של קבצים שימושיים

הפקודה - FASTOPEN

הזמן לפתיחת קובץ, סגירתו ואיתור הנתונים השייכים לו נמשך זמן רב. בייחוד אם מבנה הספריות על גבי הדיסק מורכב ורב. הפקודה מאפשרת למערכת ההפעלה לשמור את כל הפרטים על המקום הפיסי בדיסק שבו נמצאים הקבצים ואת פרטיהם ובכך לחסוך זמן רב בחיפוש. מידע זה נמצא בנפרד ב-FAT וב-Directory.
בכל פתיחה של קובץ חדש במערכת, התכנית תשמור בזיכרון RAM את כל המידע הנדרש לאיתור הקובץ בדיסק ובכך תחסוך זמן כאשר נבקש לפנות שוב לקובץ זה.

מקום הפקודה - יש להכניס את הפקודה לקובץ CONFIG.SYS, או לרשום אותה בסימן המנחה של DOS.

מבנה הפקודה -

בקובץ `INSTALL=FASTOPEN [drive:[=N]]/X:CONFIG.SYS`

בסימן מנחה `FASTOPEN [drive:[=N]]/X:DOS`

כאשר:

drive: - שם הדיסק שבו יש לאתר את הקבצים.

N - מספר הקבצים שיוגדרו לאיתור מהיר. ערך אפשרי 1-999. ברירת המחדל של מערכת ההפעלה היא 10 קבצים לאיתור מהיר.

/X - הזיכרון עבור שמירת המידע יוקצה בזיכרון הרחבה Expanded.

דוגמה: `INSTALL=FASTOPEN.EXE C:50 /X`

הפקודה מציגה 50 כניסות לאיתור ופתיחה מהירים של קבצים מכונן C. הפקודה מוכנסת בצורה זו לקובץ CONFIG.SYS.

הערות:

הפקודה ניתנת להפעלה פעם אחת בלבד ועל כן יש לבצע BOOT למערכת כדי להפעיל אותה שוב עם פרמטרים שונים. הפקודה מתייחסת לדיסקים קשיחים בלבד ופועלת אך ורק על קבצים שכבר פתוחים ופועלים, ואינה שומרת מידע על קבצים שטרם נעשה בהם שימוש כלשהו. כל כניסה לקובץ תופסת כ-48 בתים בזיכרון המחשב.

5.10.3 שינוי מספר אפשרי של קבצים פתוחים

הפקודה - FILES

הפקודה מאפשרת להגדיל את מספר הקבצים שתכנית אחת יכולה לפתוח במקביל. פתיחת קובץ לצורכי קריאה וכתיבה מחייבת את מערכת ההפעלה לשמור מידע על פעולה זו ולכן היא מגבילה את עצמה למספר קבצים קבוע על פי ברירת המחדל שלה. אם גודל זה לא מספיק, ניתן באמצעות הפקודה, FILES להתיר פתיחה של מספר קבצים גדול יותר במהלך העבודה.

מקום הפקודה - יש לרשום את הפקודה בקובץ CONFIG.SYS.

מבנה הפקודה - `FILES=n`

כאשר:

n - מספר הקבצים. ערכים אפשריים בין 8-255. ברירת המחדל של מערכת ההפעלה היא 8.

דוגמה: `FILES=30`

הפקודה מאפשרת לתכנית לפתוח 30 קבצים במקביל.

5.10.4 הגדרת תכנית זיכרון מטמון לדיסק - Disk Cache

התכנית - SMARTDRV.SYS

מקום התכנית - יש לרשום את שם התכנית בקובץ CONFIG.SYS.

מבנה הפקודה - DEVICE=SMARTDRV.SYS[גודל] [/A]

כאשר:

גודל - קיבולת זיכרון, בכפולות של 1024 בתים, שיש להקצות לזיכרון המטמון. אם לא נגדיר ערך זה, תשתמש התכנית ב-256KByte מזיכרון Extended, או בכל זיכרון ה-Expanded.

/A - אם רוצים להשתמש בזיכרון הרחבה מסוג Expanded, או כאשר רוצים להשתמש בזיכרון הרחבה מסוג Extended כזיכרון הרחבה מסוג Expanded.

דוגמה: DEVICE=SMARTDRV.SYS 512

נכתוב את הפקודה בקובץ CONFIG.SYS. היא תשתמש ב-512KByte מזיכרון ההרחבה Extended עבור Disk Cache.

הערות:

* יש לזכור תמיד את מחיר החיפוש בזיכרון המטמון. במבט ראשון נראה כי ככל שנגדיר יותר מקום עבור זיכרון המטמון נשפר את ביצועי המחשב. לחיפוש גזרה במטמון יש מחיר של זמן. אם הזיכרון המוקצה גדול מאוד והגזרה אינה נמצאת בו, מהיר יותר יהיה להביא אותה ישירות מהדיסק ולא לחפש בזיכרון. כדאי לבדוק את תגובת המערכת לגודלי מטמון שונים ועל פי בדיקות אלו להגדיר גודל מתאים למערכת המחשב.

* גירסה 4 ומעלה של DOS בלבד. זיכרון הרחבה Expanded/Extended במחשב הכרחי להרצת הפקודה. ישנן תכניות חיצוניות ל-DOS המאפשרות שימוש בזיכרון עד 640KByte לצורכי Cache כמו תכנית NCACHE-F של NORTON.

5.10.5 הגדרת דיסק בפועל - VDISK RAM

התכנית - VDISK.SYS

מקום התכנית - יש להכניס את הפקודה לקובץ CONFIG.SYS.

מבנה הפקודה בקובץ CONFIG.SYS:

עד גירסה 3.3 של DOS,

DEVICE=VDISK.SYS[גודל דיסק][/E]

או

DEVICE=VDISK.SYS[גודל דיסק][/A]

בגירסה 4 ומעלה,

DEVICE=RAMDRIVE.SYS[גודל דיסק][גודל גזרה][מספר קבצים][/E]

או

DEVICE=RAMDRIVE.SYS[גודל דיסק][גודל גזרה][מספר קבצים] [/A]

כאשר:

גודל דיסק - גודל זיכרון RAM שיש להקצות לדיסק בפועל, בכפולות של 1024 בתים. ברירת מחדל היא 64KByte וגודל מינימלי הוא 16KByte.

גודל גזרה - מספר בתים שתכיל כל גזרה בדיסק (כמה בתים מה-RAM מהווים גזרה). גדלים אפשריים: 128, 256, 512, 1024. ברירת המחדל היא 512 בתים.

מספר קבצים - כמה מקום יש להקצות לספריית השורש לצורך יצירת קבצים. מספר הקבצים הוא בכפולות של גודל הגזרה שהגדרנו חלקי 32 והתכנית תעגל את המספר שנבקש לכפולה הקרובה ביותר. הדבר נובע מכך שכל כניסה לקובץ בספריה היא בת 32 בתים. למשל, עבור גזרה בגודל 256 הכפולה היא 8. אם נבקש 41 כניסות התכנית תעגל זאת ל-48 כניסות שזו כפולת ה-8 הקרובה ביותר למעלה מ-41.

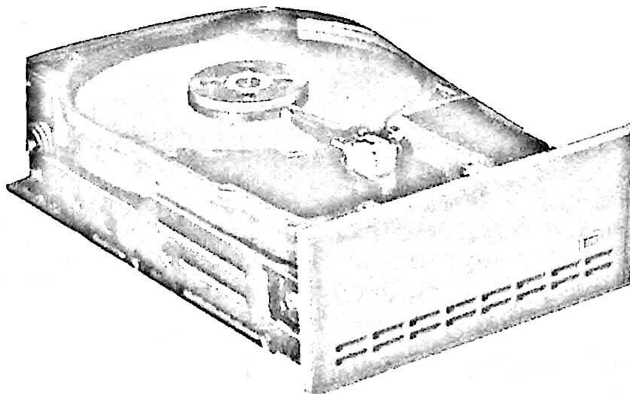
/E - שימוש באופציה זו יחייב את התכנית להשתמש בזיכרון ההרחבה (Extended) לצורך הדיסק בפועל.

/A - שימוש באופציה זו יחייב את התכנית להשתמש בזיכרון ההרחבה (Expanded) לצורך הדיסק בפועל.

הערה: לא ניתן להשתמש בשתי האופציות /E ו-/A ביחד!

דוגמה: DEVICE=RAMDRIVE.SYS 128 1024 64 /E

הפקודה תיצור דיסק בפועל (Virtual) בגודל של 128KByte, גודל גזרה של 1024 בתים ו-64 כניסות לקבצים בספריית השורש. התכנית תשתמש בזיכרון ההרחבה (Extended) לצורך הדיסק.



פרק 6

הפעלת תכניות תחת מערכת ההפעלה DOS

מערכת ההפעלה מיוצגת עבור המשתמש באמצעות **מעבד הפקודות** COMMAND.COM. אחד התפקידים של מעבד הפקודות הוא לקלוט הוראה מהמשתמש ולטפל בביצוע נאות שלה (ראה גם פרק 3). כאשר מוצג בפני המשתמש **הסימן המנחה** (Prompt), הוא יכול לבקש ממעבד הפקודות לבצע עבורו שירות כלשהו של מערכת ההפעלה. עבור כל בקשת שירות של המשתמש מבצע מעבד הפקודות סדרת פעולות לפי סדר קבוע מראש, אשר מכוונת למילוי בקשה זו באופן המתאים ביותר.

6.1 תהליך ביצוע פקודת משתמש

מעבד הפקודות קורא הוראה מלוח המקשים ומנסה למלא אותה על פי התהליך הבא:

א. מחפש באוסף התכניות הפנימיות של מערכת ההפעלה תכנית פנימית ששמה מתאים להוראה של המשתמש.
ב. מחפש קובץ חיצוני למערכת ההפעלה עם סיומת COM ושם הוראה שביקש המשתמש.
ג. מחפש קובץ חיצוני למערכת ההפעלה עם הסיומת EXE ושם הוראה שביקש המשתמש.
ד. מחפש קובץ חיצוני למערכת ההפעלה עם הסיומת BAT ושם הוראה שביקש המשתמש.

תכניות פנימיות של מערכת ההפעלה DOS טעונות דרך קבע בזיכרון המחשב ועל כן הפעלתן מהירה ביותר. מעבד הפקודות מעביר את השליטה לתכנית הפנימית. בגמר ביצוע התכנית, חוזרת השליטה למעבד הפקודות והמשתמש יכול לבקש שירותים נוספים ממערכת ההפעלה.

תהליך החיפוש והביצוע של תכניות חיצוניות למערכת ההפעלה מורכב יותר. מעבד הפקודות מחפש תכנית מתאימה על פי סדר קבוע. התכנית שתתבצע תהיה התכנית הראשונה שתתאים בצורה מלאה ומתאימה לבקשת המשתמש. חיפוש של תכנית חיצונית ייעשה תחילה בכונן ובספריה שבה נמצא המשתמש (ברירת המחדל), ורק לאחר מכן בכל אחת מהספריות המוגדרות **במסלול החיפוש** (Path). במידה ונמצא קובץ העונה לבקשה, החיפוש ייעצר והתכנית תוטען לזיכרון ותבוצע. אם לא נמצא קובץ מתאים, המשתמש יקבל הודעת שגיאה: .Bad Command or File Name

בכל ספריה שבה יתבצע החיפוש, יחפש מעבד הפקודות תחילה בין הקבצים בעלי סיומת COM, לאחר מכן בין הקבצים בעלי סיומת EXE ורק אחר כך – קבצים בעלי סיומת BAT. כלומר, אם משתמש נמצא בספריה שקיימים בה הקבצים SAMPLE.EXE, SAMPLE.COM, SAMPLE.BAT והוא רוצה לבצע את הפקודה SAMPLE, יטען מעבד הפקודות את התכנית SAMPLE.COM לזיכרון המחשב ולא את אחת התכניות האחרות בשם SAMPLE.

6.2 סוגים של תכניות חיצוניות

קיימים מספר סוגים של תכניות חיצוניות למערכת ההפעלה שניתן לטעון אותן לזיכרון המחשב ולבצע:

1. **קבצים עם סיומת COM או EXE** – קבצים אלה מכילים קוד בשפת מכונה של המעבד (CPU). על מנת לבצע תכנית עם סיומת COM, או עם סיומת EXE, יש לטעון אותה לזיכרון ולאפשר למעבד לבצע את ההוראות הרשומות בה. הצגה של תוכן קובצי שפת מכונה על גבי המסך או על גבי נייר היא חסרת משמעות למשתמש. המעוניין בכך יוכל לראות תווים שונים חסרי משמעות ואף לשמוע צפצופים צורמניים. אפשר לראות את תוכן הקבצים בייצוג הקסה-דצימלי באמצעות תוכנת שירות חיצונית כמו Norton, או PC-Tools למשל.
קובצי שפת מכונה נוצרים באמצעות שפות אסמבלר ומהדרים (Compilers) המתרגמים תכניות לקוד שפת מכונה המוכר למעבד (CPU).

2. **קבצים עם סיומת BAT**. קבצים אלה הינם קובצי טקסט המכילים הוראות של מערכת ההפעלה DOS המובנות למעבד הפקודות COMMAND.COM. קבצים עם סיומת BAT נקראים **קובצי אצווה** (Batch), מכיון שהם מכילים אוסף פקודות אשר מעבד הפקודות "יודע" לבצע, כאילו ניתנו לו מהמקלדת. ניתן להציג את תוכנם של קובצי אצווה הן על גבי המסך והן על גבי המדפסת. כל משתמש יכול ליצור קובצי אצווה (ראה הסברים על קובץ AUTOEXEC.BAT בפרק 3) ולהבין את מרבית הפקודות הרשומות בקובצי אצווה שיצרו משתמשים אחרים.

נזכיר שני סוגי קבצי תכניות נוספים שמערכת ההפעלה מטפלת בהם:
1. קובץ CONFIG.SYS - קובץ אצווה להכוונת מערכת ההפעלה (ראה פרק 3)
2. קובצי הפעלה להתקנים - קובצי תכניות (Device Drivers) המפעילות שהוסברו התקנים לא סטנדרטיים (ראה פרק 3).

6.3 ביצוע של תכניות בשפת מכונה

נניח שבדיסק, או בדיסקט קיים קובץ המכיל פקודות בשפת מכונה של המעבד, ואנו מעוניינים שהמעבד יבצע את הפקודות הרשומות בקובץ. הפתרון הבסיסי הנראה לעין הוא להעתיק את תוכן הקובץ לזיכרון, לעדכן את אוגר IP לתחילת קטע הזיכרון שבו נמצאת התכנית ולגרום להפעלה. הפתרון המתואר מיושם באופן בסיסי זה, אבל תחילה יש להשיב על שתי שאלות:

1. היכן בזיכרון יש לטעון את קובץ התכנית?
2. כיצד תדע התכנית שבזיכרון להחזיר את השליטה למערכת ההפעלה בסיום הביצוע?

ננסה לבחון את הבעיה הראשונה. פקודות התכנית צריכות לדעת היכן נמצאים נתוני התכנית ומהן הכתובות של השגרות השונות שהתכנית צריכה לבצע. חלק מכתובות אלו הן יחסיות **למקטע הקוד** (Code segment), או **למקטע הנתונים** (Data segment). קיימות כתובות אחרות, שאינן יחסיות ואיתן קיים קושי - היכן הן בזיכרון וכיצד למסור מידע זה לכל מי שפונה אליהן?

כתובות יחסיות מאפשרות לפנות לפקודות, או לנתונים, ללא תלות בכתובת המוחלטת של מקטע הקוד או מקטע הנתונים בזיכרון המחשב. הכתובת היחסית תצביע תמיד מנקודת ההתחלה של אותו מקטע בזיכרון אל הכתובת הנכונה בתוך המקטע. לדוגמה, פקודה בכתובת יחסית 100Hex תפנה תמיד לכתובת המוחלטת CS:100, בהתאם לערכו של אוגר CS באותו זמן.

כתובות שאינן יחסיות מחייבות טיפול שונה. בזמן הביצוע של התכנית חייבות להימצא בקוד התכנית כל הכתובות המוחלטות שאליהן יש לפנות. כיצד ידע המהדר מהן הכתובות המוחלטות? הפתרון הפשוט אומר, שיש לטעון תמיד את תכניות המשתמש למקום קבוע בזיכרון ואז אין כל קושי מבחינת המתכנת/מהדר, כי כל הכתובות ידועות מראש. הקצאה מסוג זה נקראת הקצאה קבועה.

הבעיה בפתרון ההקצאה הקבועה נוצרת כאשר לדוגמה, תכנית א' היתה מפעילה את תכנית ב'. אם הן היו מוכנסות לאותו מקום הן היו מוחקות זו את זו. הפתרון הוא, שמערכת ההפעלה DOS תקצה את הזיכרון בצורה דינמית. על כן התכנית אינה יודעת מראש היכן היא תוטען בזיכרון בזמן ההפעלה ויש לבנות את הקובץ בצורה שתתאים להקצאה דינמית.

מהאמור לעיל מובן, שיש להוסיף טבלה לקובץ התכנית בשפת המכונה, כדי שבעזרתה תדע מערכת ההפעלה כיצד לעדכן את הכתובות הקבועות בקוד בזמן שמוקצה לתכנית מקום בזיכרון המחשב בעת הטעינה.

תכנית השירות Load & Execute

מערכת ההפעלה מזהה שהקובץ מכיל תכנית בשפת מכונה על פי הסיומת COM או EXE. מעבד הפקודות מבקש באמצעות מנגנון הפסיקה של מערכת ההפעלה DOS, לטעון את התכנית לזיכרון המחשב ולבצע אותה. תהליך הטעינה לזיכרון והביצוע של התכנית נקרא Load & Eexcute והוא נעשה באמצעות תכנית השירות EXEC של מערכת ההפעלה (פסיקת שירות 4BHex של מערכת ההפעלה).

תכנית השירות Load & Execute מבצעת מספר פעולות:

1. מוודאת כי קיים מספיק זיכרון על מנת לטעון את התכנית לזיכרון הראשי. בעבודה תחת מערכת ההפעלה DOS, כל תכנית שיש לטעון לזיכרון לשם ביצוע חייבת להמצא כולה בזיכרון המחשב. אם אין כמות זיכרון מספקת לטעינת התכנית, המשתמש מקבל הודעת שגיאה:

```
Program too big to fit in memory
```

התכנית לא תיטען לזיכרון המחשב והסימן המנחה של מעבד הפקודות יופיע על גבי המסך.

2. אם קיים מספיק מקום בזיכרון המחשב לטעינת התכנית, תכנית השירות מקצה בזיכרון שטח הנקרא PSP (Program Segment Prefix) בתחילת שטח הזיכרון המוקצב לתכנית ולאחריו היא טוענת את התכנית.

3. תכנית השירות מעבירה שליטה לתכנית שהוטענה לזיכרון ומתחילה לבצע את הפקודה הראשונה. בקבצים עם סיומת COM הכתובת ההתחלתית לתחילת ביצוע התכנית היא כתובת 100Hex, שנמצאת לאחר טבלת PSP (הסברים מפורטים על טבלת PSP בסעיף 6.1.4). בקבצים עם סיומת EXE הכתובת ההתחלתית אינו קבוע והוא רשום בתחילת הקובץ בשפת מכונה.

התכנית יכולה להשיג שטח עבודה נוסף לשימושה בזיכרון על ידי "דריסה" של חלק ממעבד הפקודות אשר טעון בחלק העליון של הזיכרון. דריסה של חלק ממעבד הפקודות נעשה על ידי תכניות הזקוקות למרחב עבודה גדול יותר ממה שזמין עבורן בזיכרון.

תכנית המסיימת את פעולתה, מחזירה שליטה למעבד הפקודות הנמצא דרך קבע בזיכרון המחשב. כאשר השליטה חוזרת אליו, מעבד הפקודות בודק אם כל חלקיו נמצאים בזיכרון. אם הוא נמצא בשלמות בזיכרון, הסימן המנחה של מערכת ההפעלה יופיע על גבי המסך וניתן להמשיך בפעילות. אם התכנית השתמשה בזיכרון שהוקצב לחלק הנייד של מעבד הפקודות, הוא יוטען לזיכרון מהדיסק/דיסקט ורק אז יופיע הסימן המנחה.

מסיבה זאת נראה שבמקרים מסוימים מופיעה הודעת שגיאה בסיום ביצוע תכנית. הודעה זו תופיע אם הוצאנו את הדיסקט שבו הקובץ COMMAND.COM, או אם לא הגדרנו היכן נמצא קובץ זה. אם הקובץ הזה נמצא במקום שונה מאשר בכונן ברירת המחדל, נוכל לציין את מיקומו באמצעות הפקודה COMSPEC בקובץ CONFIG.SYS.

6.4 תחילית מקטע התכנית – PSP (Program Segment Prefix)

תפקידה של תכנית השירות Load & Execute של מערכת ההפעלה DOS לטעון את תכנית המשתמש לזיכרון ולהתחיל בביצוע הפקודות. תכנית זו מגדירה בזיכרון את טבלת PSP (Program Segment Prefix), אשר גודלה 100Hex בתים (256 בתים). אחרי טבלה זו מופיעה הפקודה הראשונה של התכנית.

בטבלת PSP שומרת תכנית השירות מידע חיוני הדרוש להמשך הפעילות במחשב. המידע נמסר גם לשימוש תכנית המשתמש והוא מאפשר למערכת ההפעלה להחזיר שליטה לתכנית COMMAND.COM בסיום ביצוע התכנית.

תכנית המשתמש נטענת לזיכרון מיד אחרי טבלת PSP.

6.4.1 מבנה טבלת PSP

(כל הערכים בטבלה הם הקסה-דצימליים)

כתובת	עד כתובת	יעוד
0	1	קוד לביצוע פסיקה 20Hex. באמצעות פסיקה 20Hex ניתן להחזיר שליטה למעבד הפקודות. החזרת השליטה למעבד הפקודות תתבצע בצורה תקינה, בתנאי שאוגר CS מכיל את כתובת תחילת טבלת PSP ואוגר IP שווה 0. הערך בכתובות אלו יהיה תמיד CD20.
2	3	כתובת של מקטע הזיכרון הפנוי הראשון מעבר לזיכרון שהוקצה לתכנית. בדוגמה למטה, ערך זה הוא 9FC0.
4	9	שמור לשימושי מערכת ההפעלה.
A	D	CS:IP. זוהי הכתובת שבה מסתיימת תכנית המשתמש וממנה חוזרים למעבד הפקודות. בדוגמה למטה, הכתובת 2A03:0147.
E	11	CS:IP. מצביע לשגרה המטפלת במקשי Ctrl-C. כאשר לוחצים על מקשי ^C במהלך תכנית, נוצרת פסיקה המפעילה את התכנית שכתובתה רשומה באוגרים אלה. בדוגמה למטה, הכתובת היא 2A03:1056.
12	15	CS:IP. הצבעה על תת-תכנית המטפלת בשגיאות של תכנית המשתמש. כאשר חלה שגיאת מערכת בתכנית המשתמש (למשל, חלוקה באפס), תתבצע התכנית שכתובתה רשומה באוגרים אלה. בדוגמה למטה הכתובת היא 2A03:05BF.
16	5B	מרחב כתובות זה הינו לשימוש מערכת ההפעלה.
5C	7F	תוכן שני הפרמטרים הראשונים שהמשתמש הקיש בשורת הפרמטרים כאשר הפעיל את התכנית. הצורה בה נמצאים פרמטרים אלה מקבילה לטבלת בקרה לקובץ שעדיין לא נפתח (Unopened File Control Block). בדוגמה למטה הפרמטרים הם: EXAMPLE ו-OF.
80	80	מספר התווים בשורת הפרמטרים.
81	FF	פרמטרים שרשם המשתמש כאשר הפעיל את התכנית (Command Line Parameters). בדוגמה למטה קיימים 5 פרמטרים: Example of command line parameters. כל מלה במשפט מהווה פרמטר לתכנית.

טבלת PSP הבאה מפרטת את הערכים השונים המוגדרים עבור תכנית הדוגמה. תכנית הדוגמה, שממנה נלקחה טבלת PSP, הופעלה עם שורת הפרמטרים הבאה:

```
Example of command line parameters
```

6.4.2 החזרת שליטה למעבד הפקודות

תכנית המשתמש יכולה להחזיר שליטה למעבד הפקודות COMMAND.COM בכמה דרכים:

1. קריאה לפסיקת שירות 21Hex של מערכת ההפעלה כשאוגר AH שווה 4C. פסיקת שירות 4C של מערכת ההפעלה מסמנת סיום תכנית והחזרת השליטה למעבד הפקודות. כל מרחב הזיכרון שהוקצה לתכנית משתחרר ופנוי להקצאה לתכניות אחרות.

2. קריאה לפסיקת שירות 21Hex של מערכת ההפעלה כשערך האוגר AH הוא 31. פסיקת שירות 31 מסמנת סיום חוקי של תכנית המשתמש והחזרת השליטה למעבד הפקודות. התכנית נשארת כתכנית תושבת (TSR) בזיכרון המחשב ומרחב הזיכרון שהוקצה לה אינו פנוי להקצאה לתכניות אחרות.

3. חזרה רחוקה (Far Return) מתכנית המשתמש לכתובת 0 של טבלת PSP. בחזרה רחוקה מתעדכנים האוגרים CS:IP של המעבד לכתובת ההתחלתית של טבלת PSP. חובת המתכנת לעדכן את ערך האוגרים האלה.

חזרה למעבד הפקודות בדרך 1 או 2 מומלצת יותר. באופן זה ניתן לשמור על תאימות התכניות גם בגירסאות מתקדמות יותר של מערכת ההפעלה DOS שיוצאו בעתיד.

6.4.3 כיצד יודעת התכנית היכן נמצאת טבלת PSP?

התכנית צריכה לדעת מהי הכתובת ההתחלתית של טבלת PSP, מכיון שבאמצעותה היא מקבלת פרמטרים חשובים הנחוצים לה במהלך הביצוע. כאשר תכנית השירות Load & Execute מעבירה שליטה לתכנית המשתמש, היא מעבירה אליה גם את הכתובת ההתחלתית של טבלת PSP. העברת הכתובת תלויה בסוג התכנית המועטנת לזיכרון: תכנית במבנה COM, או תכנית במבנה EXE.

כאשר מעבירים שליטה לתכנית במבנה COM, כתובת המקטע של טבלת PSP מופיעה בכל אוגרי המקטע של תכנית המשתמש (SS, ES, DS, CS) ואוגר IP מקבל את הערך 100Hex. התכנית מעדכנת את אוגרי המקטע SS, ES, DS ואת אוגר המחסנית SP בתחילת התכנית על מנת שהיא תתבצע בצורה נכונה (הסברים מפורטים בסעיף 6.1.5).

כאשר מעבירים שליטה לתכנית במבנה EXE, כתובת המקטע של טבלת PSP מופיעה באוגרי המקטע ES ו-DS בלבד. לכן, תכנית במבנה EXE חייבת לעדכן את אוגרי המקטע האלה בלבד. שאר האוגרים מעודכנים על ידי תכנית השירות Load & Execute (הסברים מפורטים בסעיף 6.1.6).

```
00  CD 20 C0 9F 00 9A F0 FE 1D F0 47 01 03 2A 56 01  ..........G...V.
10  03 2A BF 05 03 2A 03 2A 01 01 01 00 02 03 FF FF  ................
20  FF FF FF FF FF FF FF FF FF FF FF FF 7E 27 30 17  ................
30  C3 2E 14 00 18 00 56 2D FF FF FF FF 00 00 00 00  ......V.........
40  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
50  CD 21 CB 00 00 00 00 00 00 00 00 00 00 45 58 41  .............EXA
60  4D 50 4C 45 20 20 20 20 00 00 00 00 00 4F 46 20  MPLE.........OF.
68  20 20 20 20 20 20 20 20 00 00 00 00 00 00 00 00  ................
80  4A 20 65 78 61 6D 70 6C 65 20 6F 66 20 63 6F 6D  J.example.of.com
90  6D 61 6E 64 20 6C 69 6E 65 20 70 61 72 61 6D 65  mand.line.parame
A0  74 65 72 73 20 20 20 20 20 20 20 20 20 20 20 20  ters............
B0  20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20  ................
C0  20 20 20 20 20 20 20 20 20 20 20 0D 00 00 00 00  ................
D0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
E0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
F0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
```

דוגמה לטבלת PSP של תכנית משתמש

6.5 קובצי שפת מכונה במבנה COM

קובצי תכניות בעלי סיומת COM מכילים שפת מכונה של המעבד. מרבית המהדרים (Compilers) והאסמבלרים (Assemblers) הקיימים כיום, אינם יוצרים קובצי שפת מכונה במבנה זה. המבנה של קובצי COM פשוט ייותר ממבנה קבצים במבנה EXE, אך הם מציבים מגבלות רבות למתכנת.

טעינת תכניות במבנה COM פשוטה ביותר. קובץ שפת מכונה במבנה COM בנוי באופן כזה שבתהליך הטעינה לזיכרון נבנה PSP ובהמשכו נטען קובץ שפת המכונה כפי שהוא, ללא כל שינוי. קובץ התכנית במבנה COM שמור על גבי הדיסק/דיסקט מוכן להרצה.

בגמר הטעינה לזיכרון מועברת השליטה אל התכנית, לכתובת 100Hex יחסית למקטע שבו מתחילה טבלת PSP. כתובת 100Hex הינה קבועה עבור כל התכניות במבנה COM ובה נמצאת הפקודה הראשונה בתכנית.

המגבלה העיקרית של תכנית במבנה COM היא הגודל שלה, אשר מוגבל לקטע זיכרון בגודל של 64KByte בלבד. גודל זה מותאם לגודל **מקטע** (Segment) בארכיטקטורה של מעבדי אינטל. תכנית המחייבת זיכרון גדול מ-64KByte, חייבת להשתמש במנגנון של רבדים (Overlays - חלקים נפרדים של התכנית הנשמרים על גבי הדיסקט ונטענים לאותו שטח זיכרון לשם ביצוע). שימוש בנוהל זה מאט את ביצוע התכנית ומסרבל את תהליך העבודה במחשב.

כיום, כשהזיכרון הראשי העומד לרשות המתכנת גדול, מהווה מגבלת הזיכרון של קובצי COM חיסרון, ולכן השימוש בקבצים אלה הולך ופוחת. עם זאת, תהליך הטעינה וההפעלה של קבצים הבנויים במבנה COM מהיר יותר מתהליך הטעינה וההפעלה של קובצי EXE.

6.6 קובצי שפת מכונה במבנה EXE

קובצי שפת מכונה במבנה EXE מורכבים יותר מאשר קובצי COM. גודל תכנית במבנה EXE אינו מוגבל למקטע יחיד בגודל 64KByte ולכן היא יכולה להכיל מספר **מקטעים של קוד** (Code Segments) ומספר **מקטעים של נתונים** (Data Segments). השימוש במספר מקטעים בתכנית מאפשר למתכנת ליצור תכניות מורכבות יותר ומתאימות יותר לצרכים העכשוויים שלו.

על מנת שתכנית במבנה EXE תוכל להתבצע, תכנית השירות Load & Execute צריכה לעדכן מספר פרטים בקובץ שפת המכונה בכל פעם שהוא נטען לזיכרון. הכתובת ההתחלתית בתכנית שאליה מעבירה תכנית השירות את השליטה שונה מתכנית לתכנית ואינה קבועה כמו בקבצים במבנה COM. בראש קובץ (Header) שפת מכונה במבנה EXE קיימות הוראות כיצד לטעון את התכנית. הן מיועדות עבור תכנית השירות Load & Execute אשר משתמשת בהן כדי להכין את התכנית הנטענת להרצה במחשב.

"הכנה" של התכנית הנטענת לקראת הרצה כוללת בעיקר עדכוני כתובות. בכל פעם שתכנית נטענת לזיכרון, הכתובת ההתחלתית שלה משתנה בהתאם לתפוסת זיכרון המחשב ומשתנות גם הכתובות של השגרות השונות הכלולות בה.

6.6.1 מבנה כותרת (Header) של קובצי EXE

	כתובת	גודל	יעוד
.1	00Hex	2 בתים	הערך יהיה תמיד האותיות MZ (ערכים 5AHex ו-4DHex) והוא מציין שהקובץ מכיל קוד מכונה במבנה EXE.
.2	02Hex	2 בתים	מלה ברוחב 16 סיביות. הערך הקיים במלה זאת הוא שארית החלוקה של מספר הבתים הכולל בקובץ ב-512 בתים: (File Size) Modulo (512).
.3	04Hex	2 בתים	גודל הקובץ (File Size). זהו הגודל הכולל של הקובץ, כולל כותרת הקובץ (Header).
.4	06Hex	2 בתים	מספר הכניסות/מקומות בקובץ אשר תכנית Load & Execute צריכה לעדכן. מספר זה מציין כמה כתובות של מקומות עדכון בקובץ שפת המכונה קיימות בטבלת המיקום הנמצאת בכותרת הקובץ.
.5	08Hex	2 בתים	גודל כותרת הקובץ. מספר זה מציין לתכנית הטעינה כמה קטעים (Paragraphs) בתחילת הקובץ מכילים נתונים לטעינה. בכל קטע יש 16 בתים.
.6	0AHex	2 בתים (MIN)	מספר מינימלי של קטעים בזיכרון המחשב אשר דרושים לתכנית מעבר לכתובת האחרונה בזיכרון המוקצית התכנית. כל קטע (Paragraph) מכיל 16 בתים. ערך זה הוא בדרך כלל 0.

.7	0CHex	2 בתים (MAX)	מספר מירבי של קטעים בזיכרון הדרושים מעבר לכתובת האחרונה בזיכרון המוקצית לתכנית. כל קטע (Paragraph) מכיל 16 בתים. ערך זה הוא בדרך כלל 0FFFFHex. אם ערך MIN וערך MAX הם 0, תטען תכנית השירות את התכנית באיזור הגבוה ביותר האפשרי בזיכרון RAM.
.8	0EHex	2 בתים	ערך התחלתי של הכתובת שתוטען ל**מקטע המחסנית** (Stack Segment). הערך יחסי לתחילת התכנית, נמדד במספר קטעי זיכרון מתחילתה ומעודכן על ידי תכנית השירות Load & Execute לפי הכתובת ההתחלתית שבה נטענת התכנית.
.9	10Hex	2 בתים	גודל מקטע המחסנית המוקצה לתכנית. ערך זה יש לטעון ל**אוגר מצביע המחסנית** (Stack Pointer). ערך זה יחסי למקטע המחסנית ואינו דורש עדכון על פי כתובת התחלתית של התכנית.
.10	12Hex	2 בתים	ערך זה נועד לבדיקת ואימות תוכן קובץ שפת המכונה. מסכמים את הערכים של כל הבתים בקובץ, מחלקים ב-65536 ואת השארית שומרים ב-2 בתים אלה. ערך זה נקרא Checksum.
.11	14Hex	2 בתים	ערכו של אוגר **מצביע ההוראות** (Instruction Pointer). ערך זה יוכנס לתוך אוגר IP כאשר השליטה תועבר לתכנית. הוא איננו משתנה או מעודכן על ידי תכנית השירות Load & Execute על פי מקום תחילת התכנית בזיכרון.
.12	16Hex	2 בתים	ערך התחלתי של **מקטע קוד התכנית** (Code Segment). ערך זה מעודכן על ידי תכנית השירות Load & Execute בזמן הטעינה לזיכרון.
.13	18Hex	2 בתים	מקום טבלת הכתובות המחייבות עדכון בזמן הטעינה. מקום הטבלה יחסי לתחילת הקובץ בשפת המכונה ומספר הכניסות בה נקבע על פי מספר העדכונים שיש לערוך בו (המספר נמצא בכתובות 06-07 של כותרת הקובץ). עבור כל כניסה בטבלה, תכנית השירות Load & Execute ניגשת לכתובת מתאימה של התכנית בזיכרון ומוסיפה לערך הקיים בה את ערך מקטע הקוד ההתחלתי שאליו נטענה התכנית. הוספת הערך לכל המקומות המופיעים בטבלה מבטיחה עדכון של כל הכתובות, על מנת שהתכנית תתבצע כראוי. ראה את הדוגמה.
.14	1AHex	2 בתים	המספר הסדורי של רובד התכנית (Overlay Number). כאשר התכנית בנויה ממספר **רבדים** (Overlays) ניתן לכל קובץ מספר. מספרה של התכנית הראשית הוא 0. ערך זה יהיה 0 עבור תכנית ראשית ללא חלקים.

6.6.2 תכנית EXE לדוגמה

תכנית הדוגמה (ראה גם קובצי *.KEYPRESS בדיסקט תכניות הדוגמה) קוראת מהמשתמש לחיצות מקשים. היא אוגרת את ערך המקשים שנלחצו עד שנלחץ מקש Enter, או עד שהוקשו 16 מקשים. בשלב זה התכנית לא רושמת כל נתון שהוא על גבי המסך, אך בגמר הכנסת הנתונים היא מציגה על גבי המסך בסדר הפוך את כל הערכים שאגרה.

תכנית הדוגמה הינה בשפת אסמבלי. היא תורגמה בעזרת אסמבלר, עברה תהליך קישור (Link) שבסופו התקבל קובץ במבנה EXE. התכנית מוצגת כפי שהופקה מתכנית האסמבלר: בכל שורה מופיע מספר השורה, כתובת הפקודה בתכנית, קוד שפת המכונה והפקודה בשפת האסמבלי.

הבהרות:

* בכל מקום שבו מופיע הסימן R ---- הכוונה לכתובת יחסית לתחילת המקטע. המקום במקטע משתנה על פי הכתובת ההתחלתית שבה נטענת התכנית בזיכרון.

* כל הערכים הם הקסה-דצימליים.

* בתכנית הדוגמה יש צורך לעדכן רק כתובת אחת בקובץ (2 בתים בכתובות 5 ו-6 במקטע הקוד) לפני העברת השליטה לתכנית. כתובת זו היא כתובת מקטע הנתונים (שליד כתובתו מופיע הסימן R ----). בכתובת 1EHex בתחילת הקובץ נמצא שתי מלים המייצגות כתובת היסט שערכה 0005 ומקטע שערכו 0021. כלומר, שבטעינת התכנית צריכה תכנית השירות Load & Execute לעדכן את כתובת מקטע הנתונים בהתאם לכתובת הזיכרון המוקצית לו. בזמן הטעינה תיגש תכנית השירות לכתובת 0021:0005 בקובץ שפת המכונה שנטען לזיכרון ותטען בה את ערך מקטע הנתונים.

התכנית:

```
Addr  Value/Opcode           Command

0000                         DSEG SEGMENT              ;מקטע מידע
0000  0010[                    LINE DB 10H DUP(0)
            00
                  ]
0010                         DSEG ENDS

0000                         SSEG SEGMENT STACK        ;מקטע מחסנית
0000  0100[                     DW 100H DUP(?)
            ????
                  ]
0200                         SSEG ENDS

0000                         CSEG SEGMENT              ;מקטע קוד
                                ASSUME CS:CSEG,DS:DSEG,SS:SSEG
```

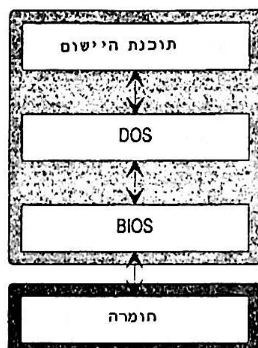
```
Addr  Value/Opcode          Command        (המשך)

0000                        EXAMPLE PROC FAR
0000  1E                       PUSH DS            ;כתובת חזרה
0001  2B C0                    SUB AX,AX          ;      DOS-ל
0003  50                       PUSH AX

                                                  כתובות 5,6
0004  B8 ---- R                MOV AX,DSEG        ;DS איתחול
0007  8E D8                    MOV DS,AX

0009  B3 00                    MOV BL,0
000B  8D 36 0000 R             LEA SI,LINE
000F  B9 0010                  MOV CX,10H
0012                        READ-LOOP:            ;קריאת מקשים
0012  B4 00                    MOV AH,0
0014  CD 16                    INT 16H
0016  FE C3                    INC BL
0018  3C 0D                    CMP AL,0DH          ;? CR האם
001A  74 05                    JZ WRITE-LOOP
001C  88 04                    MOV [SI],AL         ;שמור מקש
001E  46                       INC SI
001F  E2 F1                    LOOP READ-LOOP

0021                        WRITE-LOOP:           ;לולאת הדפסה
0021  B4 0E                    MOV AH,0EH         ;קוד הדפסה
0023  4E                       DEC SI
0024  8A 04                    MOV AL,[SI]
0026  CD 10                    INT 10H            ;BIOS פסיקת
0028  FE CB                    DEC BL
002A  75 F5                    JNZ WRITE-LOOP
002C  CB                       RET
                            EXAMPLE ENDP
002D                        CSEG ENDS
                               END EXAMPLE
```



מבנה כותרת (Header) של קובץ הדוגמה

כתובת יחסית לתחילת קובץ	ערך Hex בית/מלה	יעוד
00Hex	4D 5A	סימון קובץ שפת מכונה במבנה EXE. הצירוף יהיה תמיד MZ בערך תווי.
02Hex	003D	שארית החלוקה של גודל התכנית בערך 512. במקרה זה ערך זה שווה לגודל קוד התכנית, מכיון שגודל התכנית קטן מ-512 בתים (3DHex בתים).
04Hex	0003	גודל התכנית הוא 3 קטעים (48 בתים) מעגלים את גודל התכנית לכפולה של 16 אף על פי שגודלה 3DHex.
06Hex	0001	מספר איברים בטבלת המיקום.
08Hex	0020	מספר הקטעים השייכים ל-Header. סה"כ 512 בתים = 16*20Hex
0AHex	0000	מספר הקטעים (MIN) בזיכרון הדרושים לתכנית מעבר למה שהוקצה בטעינה.
0CHex	FFFF	מספר הקטעים (MAX) הדרושים לתכנית מעבר למה שהוקצה בטעינה.
0EHex	0001	מקום המחסנית בתוך התכנית. נמדד בקטעים בגודל 16 בתים.
10Hex	0020	ערך התחלתי של מצביע המחסנית. ערכו 200 בתים כי הגדרנו 100 מלים.
12Hex	EEA6	ערך Checksum של הקובץ.
14Hex	0000	ערך התחלתי שיוכנס לאוגר IP כאשר התכנית תקבל שליטה.
16Hex	0021	מקום תחילת מקטע הקוד בקובץ. בדוגמה תופש ה-Header 20 קטעים ומקטע הנתונים תופס קטע אחד, ולכן מקטע הקוד מתחיל לאחר 21 קטעים.
18Hex	001E	מקום טבלת מיקום הקבצים.
1AHex	0000	מספר סידורי של תכנית (Overlay). <<< ראה המשך...

כתובת יחסית לתחילת קובץ	ערך Hex בית/מלה	יעוד
<<< המשך		
1EHex	0005 0021	כתובת היסט. כתובת מקטע. שתי מלים אלו מגדירות איזה מקום בקובץ יש לשנות במהלך הטעינה. בדוגמה יש לשנות מקום אחד בלבד (ערך מקטע קוד) והמקום אותו יש לעדכן אכן נמצא בכתובת 0021:0005 מתחילת הקובץ. בתכנית הדוגמה יש לעדכן רק כתובת אחת. בתכניות אחרות יתכן שיעודכנו כתובות רבות.

Name	Size	Align	Combine
CSEG	002D	PARA	NONE
DSEG	0010	PARA	NONE
SSEG	0200	PARA	STACK

גודלי המקטעים השונים

פרק 7
וירוסים במחשב האישי

השוואות בין תכונות מחשבים לבין מערכות ביולוגיות מתחום החי שייכות יותר לתחום הפילוסופיה של עיבוד נתונים ובקרה, ולכן לא נעסוק בהן בספר זה. בכל זאת, כאשר מתייחסים לוירוסים במחשב חייבים להביא את המקבילה הביולוגית על מנת להבהיר את מקור השם.

7.1 מה עושה הוירוס?

וירוס המחשב קיבל את שמו בשל צורת ההתפשטות וההידבקות ממנו, אשר מזכירה את הוירוסים הפוגעים בבריאותנו. הוירוס הביולוגי הוא קטע של קוד גנטי (DNA) רדום שנמצא בטבע. כאשר הוא מתעורר, הוא מחפש תא חי, תוקף וחודר לגרעין שלו. הוירוס מחפש בתוך הקוד הגנטי מקום מסוים שבו הוא מסוגל לפעול. הוא חותך פיסית את הקוד הגנטי ומכניס את עצמו במקום החתך (גם הוירוס הוא קטע של קוד גנטי). כך, הרצף של הקוד הגנטי נשמר, אבל נוסף לקוד הבסיסי קיים בקוד הגנטי החדש קטע של קוד גנטי חדש המייצג את הוירוס. קטע הקוד החדש הזה משנה את תפקוד התא החי. הקוד החדש מורה לחלבוני התא ליצר וירוסים חדשים, אשר יוצאים לדרכם וכך מתפשטת המגפה בצורה אקספוננציאלית.

וירוס המחשב זהה בפעולתו לוירוס הביולוגי, אלא שבמקום תכנית גנטית ממנה מורכב הוירוס הביולוגי, הוא מורכב מתכנית מחשב. המיתוסים שנוצרו סביב הוירוסים במחשב אינם נכונים ברובם. ההכרה כי **הוירוס היא תכנית מחשב לכל דבר** והבנת הרעיון המרכזי בסוג תכניות זה, יבהירו את יכולתו. כמו בוירוס הביולוגי, כך גם כאן אין אפשרות לחיסול מוחלט של הוירוס, כי עבור כל תכנית המסוגלת לטפל ולחסל וירוסים, ימצא מי שיבנה תכנית וירוס שעוקפת את תכנית המניעה.

כדי להבהיר נקודה זאת, תמצא בנספח תכנית וירוס פרמיטיבית (**שאינה מזיקה!**). הסברנו באופן מדויק את דרך פעולתה וכיצד אפשר להסיר אותה. תכנית הדוגמה, על אף היותה פשוטה ולא מתוחכמת, אינה ניתנת לגילוי (לפי מיטב ידיעתנו) על ידי אף לא אחת מתכניות האנטי וירוס הקיימות היום בשוק. התלבטנו רבות בשאלה אם להציג תכנית דוגמה לוירוס, מכיון שתכנית כזו עלולה לשמש קרש קפיצה למחבלים למיניהם. הפשרה שקיבלנו היא להדגים את הנושא בעזרת שיטה פרמיטיבית, כך שניתן להתגונן מפניה באמצעים פשוטים ללא צורך בתכניות אנטי וירוס מיוחדות.

בוירוס מחשב ניתן לטפל. קיימות תרופות לטיפול ומניעה של הוירוס וקל לפתח תכניות לטיפול בוירוסים חדשים, בניגוד לאפשרויות ההגנה מפני וירוסים הפוגעים בבריאות. יש לזכור כי **הוירוס התוקף את המחשב הינו תכנית מחשב!** אך בניגוד לתכניות מחשב יישומיות שנועדו ליצור דברים לתועלת המשתמש, כל כוונתו של מתכנת הוירוסים היתה לפגוע במידע האגור במחשב, להאיט את פעולתו ולגרום אך ורק נזקים!

ישנם סוגים רבים של וירוסים הנבדלים זה מזה בסוג הנזק שהם גורמים, במידת הנזק, בצורת ההתפשטות ובמבנה. עם זאת, מרכיבים משותפים שנמצא ברוב הוירוסים מאפשרים להלחם בהם כתופעה. המקור העיקרי להתפשטות הוירוסים הינו העתקות התוכנה מסיביות ושימוש בעותקים לא חוקיים של תכניות מחשב, שלא תמיד ידוע מהיכן הגיעו ומה כלול בהן.

הנשק העיקרי במלחמה נגד הוירוסים הינו בהבנת דרך פעולתם, מהותם ודרכי התפשטותם. הבנה זו מאפשרת למנוע הידבקות בוירוסים במחשב ולטפל בהם לאחר שתקפו את המחשב. הוירוסים גורמים נזק בעיקר למערכת התוכנה והנתונים שבמחשב. עיקר ההשקעה במערכת המחשב הינו בנתונים האגורים בו, ולא בחומרה המרכיבה את המערכת. תוכנה ונתונים הינם המרכיב העיקרי בעלויות, ואילו החומרה הינה מרכיב שולי. לעתים לא ניתן להעריך בכסף את ההשקעה הרבה ביצירת מאגרי המידע, שלב שנמשך לעתים שנות עבודה רבות. ידוע על וירוסים הגורמים נזק עקיף לחומרה במחשב, כמו למשל וירוס הגורם לקריאה בלתי פוסקת ורציפה מגזרת BOOT בדיסק/דיסקט ובכך הוא מקצר את אורך חייו.

7.2 מהו וירוס מחשב וכיצד הוא פועל?

וירוס הוא תכנית מחשב שנמצאת על גבי דיסקט/דיסק וכך ניתן למעשה להעבירו ממערכת מחשב אחת לחברתה. מכיון שכך, כדי שהוא יוכל לחדור למערכת המחשב יש צורך לבצע את הפקודות הרשומות בתכנית הוירוס. התכנית המרכיבה את הוירוס בנויה בדרך כלל משלושה שלבים:

א. **שכפול** - תכנית הוירוס מעתיקה את עצמה לתוך זיכרון RAM של המחשב ולתוך הדיסק/דיסקטים במערכת. תכנית הוירוס נמצאת בדרך כלל כתכנית תושבת (TSR) בזיכרון המחשב.

ב. **הסתתרות** - תכנית הוירוס קובעת להיכן תוטען בזיכרון ומהיכן תפעל.

ג. **חבלה** - תכנית הוירוס מבצעת פעילות כלשהי על התקני המחשב. לדוגמה, פורמט של הדיסק הקשיח, או ציור כדור קופצני על המסך. חשוב לזכור:

1) שלב החבלה אינו מתבצע בכל פעם שהתכנית נטענת לזיכרון המחשב ואם מתבצעת חבלה, לרוב אין היא מיידית. הוירוס מחכה לעתים לביצוע פעולה מסוימת, או לתאריך/שעה מסוימים, על מנת לפעול ולחבל (זכור את תאריך יום שישי ה-13 בחודש).

2) ישנם וירוסים שמיועדים רק להרגיז, אך לא לפגוע בפעולת המחשב. עם זאת, גם וירוסים כאלה עלולים לפגוע בנתונים.

שלב א' הוא השלב היחיד ההכרחי לקיומו של וירוס. שלבים ב' ו-ג' אומנם קיימים בכל הוירוסים שיש היום, אבל הם אינם הכרחים. בוירוס שהכנסנו בתכנית הדוגמה קיימים שלב א' ושלב ג'. שלב ג' בדוגמה שלנו כותב הודעה על המסך בלבד ואינו פוגע במידע. התפשטות הוירוסים מהירה ביותר ככל שגדל ההיקף של העתקה בלתי חוקית של תוכנה.

ישנם וירוסים הבנויים משני מרכיבים (כמו זכר ונקבה). חלק אחד הוא התכנית הראשית וחלק שני הוא תכנית הפעלה שלה. התכנית הראשית מוחבאת בקובץ נתונים, או באיזור כלשהו בדיסק או בדיסקט. תכנית ההפעלה מוצאת את הוירוס הראשי ומפעילה אותו. על מנת שוירוס זה יוכל להתפשט ולפעול, הוא זקוק לתכנית הראשית שלו.

7.3 היכן נמצא הוירוס?

מחשב כבוי אינו מכיל בזיכרון הראשי כל וירוס! לוירוס אין יכולת להדביק את התקני החומרה במחשב, או את רכיבי ROM BIOS, מכיון שזיכרון זה הוא זיכרון קריאה בלבד. הוירוס אינו יכול להדביק דרך קבע את זיכרון RAM הראשי, מכיון שתוכנו נמחק כאשר מפסיקים את המתח. יוצא מכך שהוירוס במחשב קיים רק על גבי המדיה המגנטית (דיסק קשיח, או דיסקטים) והוא נכנס לזיכרון המחשב ומתחיל לפעול בו רק כאשר נפעיל את המחשב ונבצע את תהליך BOOT.

את הוירוסים ניתן למצוא במספר מקומות אופייניים בזיכרון:

* **גזרת BOOT** - באחד משלבי תהליך BOOT, תכנית ROM BIOS מבצעת את הקוד הרשום בגיזרת BOOT וטוענת את מערכת ההפעלה. מכאן, שהמקום האופייני והיעיל ביותר מבחינת הוירוס, הוא לשכפל את עצמו על גזרה זו. הימצאות וירוס בגזרת BOOT מבטיחה שהוא יוטען לזיכרון המחשב בעת ההפעלה שלו ובכך יוכל להדביק את זיכרון המחשב מיד (כמו וירוס פינג-פונג למשל). וירוס כזה מכיל קוד הדבקה המאפשר לו לשנות את גזרת BOOT של כל דיסק/דיסקט הקיים במערכת, או של דיסקטים שיוכנסו לשימוש במהלך העבודה.

* **קובצי מכונה** - הוירוס משכפל את עצמו בתחילת קובץ המכיל קוד בשפת מכונה. קובצי שפת מכונה הם קבצים עם סיומת EXE, COM, SYS, BIN, קובצי Device Drivers לטיפול בהתקנים, או כל סוג אחר של תכניות שנטענות למחשב. קובצי נתונים, קובצי טכסט של מעבדי תמלילים וכדומה מכילים נתונים בלבד ולא קוד תכנית לביצוע ולכן אינם יעד מתאים.

כאשר המשתמש מבקש לבצע תכנית בשפת מכונה, הקוד מתבצע מיד עם טעינת התכנית לזיכרון הראשי. ביצוע קוד התכנית על ידי המעבד, הוא תנאי הכרחי לכניסת הוירוס למערכת. וירוסים הפוגעים ומשכפלים את עצמם על קובצי שפת מכונה, מנפחים את גודלו של הקובץ ומשנים אותו ולכן קל לאתר אותם בקלות יחסית, אם ידוע הגודל המקורי של הקובץ.

כאשר מריצים תכנית נגועה בוירוס, מתבצעת תחילה תכנית הוירוס שחדרה לתחילת התכנית, ורק לאחר מכן מועברת השליטה לקוד התכנית. תהליך זה שקוף למשתמש וקשה לראות ולהבחין בכניסת הוירוס למחשב.

* **קובצי נתונים** - הוירוס משכפל עותק של עצמו בתחילת או בסוף קובץ הנתונים. לא ניתן להידבק מוירוס הנמצא על קובץ נתונים מכיון שמערכת ההפעלה אינה מבצעת קבצים אלה. אולם וירוס הנדבק לקובץ נתונים גורם בחלק מהמקרים לאובדן ונזק, מכיון שהוא משנה את הקובץ. שינוי מבנה הקובץ גורם לכך שתכניות יישום המטפלות בקבצים לא יוכלו למצוא את הנתונים האגורים בהם, או שיטפלו בנתונים משובשים.

7.4 הידבקות מוירוסים ודרכי התפשטותם

הוירוסים אינם מתפשטים דרך האויר, אלא באמצעות מדיה מגנטית או רשתות תקשורת בין מחשבים. הם גם אינם צריכים יד אדם מכוונת כדי להתפשט. מחשב לא יכול להידבק, אלא אם העתיקו לתוכו את הוירוס והריצו בו תכנית נגועה בוירוס. מכאן, צריך לבדוק את המקרים השונים בהם קיים מגע בין מערכת הדיסקים במחשב האישי לבין מדיה מגנטית אחרת המכילה וירוסים.

האפשרות הנפוצה להידבק מוירוס קיימת כאשר מכניסים דיסקט נגוע בוירוס לכונן הדיסקטים במחשב ומפעילים אותו. הדיסקט הנגוע מדביק את הזיכרון ואת הדיסק הקשיח.

אפשרות נוספת, אשר הולכת ונפוצה יותר היא באמצעות רשת תקשורת, אשר קושרת מחשבים רבים בסביבת עבודה אחת. תכנית וירוס יכולה לכתוב נתונים על דיסק/דיסקט של משתמש מרוחק, או שהמשתמש יכול לקרוא נתונים מדיסק או דיסקט אשר נגועים בוירוס.

מצב שבו קיימים וירוסים במערכת המחשב נכנה **"מערכת מחשב נגועה"**, אשר הופכת בעצמה ליצרנית וירוסים. מערכת נגועה הפועלת ללא דיסק קשיח מייצרת וירוסים עד שנכבה אותה. היא תמשיך לייצר וירוסים בהפעלה הבאה רק אם נשתמש בה בדיסקטים נגועים. בדרך כלל, מערכת ללא דיסק קשיח מדביקה גם את דיסקט BOOT המפעיל אותה וברגע שנפעיל אותה מחדש עם דיסקט BOOT נגוע, היא שוב הופכת למערכת נגועה. אם נפעיל את המערכת עם דיסקט BOOT מקורי, או עם דיסקט לא נגוע, המערכת תחזור להיות **"נקיה"**. יש לזכור שדיסקטים שנפגעו בוירוס לא יהפכו ל"נקיים" אם נסיר אותם מהמחשב. בשימוש הבא הם עלולים להדביק דיסקטים נוספים.

במערכת הכוללת דיסק קשיח אשר נגוע בוירוס, המערכת תשאר נגועה עד ש"ננקה" אותה בעזרת תוכנה יעודית להסרת וירוסים. מערכת נגועה מדביקה כל דיסקט שמוכנס אליה בשלב BOOT, או תוך כדי פעילות. כאשר יודעים שהמערכת נגועה, אסור להכניס אליה דיסקטים נוספים כדי שלא נדביק בוירוס גם אותם.

יצרני הוירוסים חיקו את פעולת הוירוס הביולוגי ולכן, ברוב המקרים, אין אנו יודעים אם מערכת המחשב נגועה או לא. לחלק מוירוסי המחשבים, כמו לוירוס הביולוגי, יש זמן דגירה. בתקופה זו לא מופעלות שגרות ההרס

שלו, אלא רק שגרות ההדבקה ולכן קשה לזהות ולאתר וירוס כזה. תהליך ההדבקה והשכפול תלוי ב"דמיונו" הפורה של מתכנת הוירוס, אבל ישנן מספר דרכים אופייניות לביצוע השכפול:

א. ביצוע תהליך BOOT במחשב. כאשר בכונן A מוכנס דיסקט הנגוע בוירוס יגרום תהליך BOOT להדבקת דיסקטים נוספים והמסוכן מכל – הדבקת הדיסק הקשיח.

ב. הפעלת מחשב המכיל בשלב BOOT מדיה מגנטית שאינה נגועה בוירוס, לא תגרום להדבקת המחשב. אך אם בשלב כלשהו נבצע את הפעולות הבאות, המחשב ידבק מוירוס:

1) נריץ תכנית מחשב או קובץ אצווה הנגועים בוירוס.

2) נכניס דיסקט נגוע בוירוס לכונן הדיסקטים במערכת ונבקש לטעון ממנו תכנית כלשהי.

3) נבקש לבצע תכנית או קובץ אצווה ממדיה מגנטית אחרת הקשורה למחשב ברשת תקשורת. התפשטות דרך רשת תקשורת היא המהירה ביותר בשל ריבוי המשתמשים המחוברים לרשת בו-זמנית. כאשר תכנית הוירוס מתבצעת, הוירוס מתמקם בזיכרון המחשב ומדביק את המדיה המגנטית הקיימת במערכת. אם מעתיקים באמצעות רשת התקשורת תכנית נגועה בוירוס, אך לא טוענים אותה לזיכרון, הוירוס אינו פעיל. כאשר נטען את התכנית הנגועה הוירוס יכנס גם הוא לזיכרון ויהפך לפעיל. יש לזכור כי תכנית או קובץ נגועים אינם הופכים מיידית את המחשב לנגוע כולו בוירוס!

ג. הכנסת דיסקט שאינו נגוע בוירוס למערכת נגועה, שבה הוירוס נמצא בזיכרון המחשב, תגרום מיידית להדבקה שלו בוירוס או בוירוסים הפועלים במחשב הנגוע. תכנית הוירוס עוקבת בדרך כלל אחר מצב הדיסקטים במערכת המחשב וברגע שהיא מזהה שהכנסנו דיסקט חדש, היא משכפלת את עצמה על הדיסקט החדש. על כן, כל דיסקט שיוכנס למחשב נגוע יודבק גם הוא.

מכיון שהדיסקטים הם אמצעי ההעברה הנפוץ ביותר של מידע, הרי בכל מערכת שבה נבצע תהליך BOOT עם הדיסקט הנגוע, נגרום להדבקה מיידית של המערכת. ניתן לראות על פי שלבים אלה את ממדי ההתפשטות ומהירות ההתפשטות הגדולה של וירוסי המחשב.



7.5 דרכי טיפול בוירוסים ומניעתם

המלחמה בוירוסים יעילה יותר לאחר היכרות עם דרך פעולתם. ניתן לנקוט מספר צעדים על מנת להקטין את ממדי התופעה, או למנוע אותה כליל:

א. גיבוי נאות של התוכנה והנתונים במחשב (עיין בפרק 5 הדן בתחזוקת דיסקים). רק גיבוי מלא יציל מנזקיו של וירוס! ללא גיבוי, המערכת נותרת חשופה לפגעים ונזקים.

ב. הגנה על דיסקטים המכילים תכניות או גיבוי. דיסקט המוגן בתווית הגנה (או תריס בדיסקט "3.5) אינו מאפשר לכונן לכתוב עליו. ההגנה בפני כתיבה מושגת באמצעות מעגלי החומרה של כונן הדיסקטים והוירוס אינו יכול להסיר הגנה זאת בשום אופן. מכיון שתהליך ההדבקה כרוך בכתיבה של תכנית הוירוס על הדיסקט, הרי שהגנה זו תמנע את הדבקתו.

ג. שינוי תכונה של כל קובצי COM ו-EXE **לקריאה בלבד** (Read Only). את תכונת הקבצים אפשר לשנות על ידי תכנית ATTRIB של מערכת ההפעלה, או באמצעות תכניות שירות למיניהן כמו Norton או PC-Tools. לא כל הוירוסים יודעים לעקוף את ההגנה בפני כתיבה (ATTRIB) של הקבצים וכך מקטינים את סיכויי הפגיעה. ראה תכנית לשינוי תכונות קבצים בנספח של תכניות הדוגמה.

דוגמה לשימוש בפקודה ATTRIB

```
A>  ATTRIB +R *.EXE
A>  ATTRIB +R *.COM
```

פקודות אלו יהפכו את כל הקבצים בכונן A בעלי סיומת COM או סיומת EXE ל"קבצים לקריאה בלבד". התכונה "קריאה בלבד" אינה מאפשרת למחוק או לשנות את תוכנם של קבצים בעלי תכונה זו.

ד. ביצוע BOOT רק דרך הדיסק הקשיח במערכת שבהן הוא מותקן. ביצוע BOOT דרך כונן דיסקטים מסוכן יותר, מכיון שדיסקטים משמשים להעברת מידע בין מחשבים והסיכוי להידבק מהם גדול יותר.

קיימת אפשרות לחסום ביצוע BOOT דרך כונן A. אפשר לקבוע חסימה כזו במערכות עם דיסק קשיח וכונן דיסקטים יחיד. החסימה של כונן A נחוצה בעיקר לאלה השוכחים להוציא דיסקט מכונן זה לפני ביצוע BOOT דרך הדיסק הקשיח. דרך זו אינה מומלצת למשתמשים, אשר אינם בקיאים בחומרת המחשב.

ה. שימוש בתכניות שנועדו להילחם בוירוסים. תכניות אנטי-וירוס למיניהן הקיימות בשוק מאפשרות לבדוק דיסק או דיסקט והן מספקות למשתמש אזהרה אם הם נגועים. רוב התוכנות גם מאפשרות ניקוי של הוירוסים שנמצאו. מומלץ להשתמש בתכניות אנטי-וירוס על מנת לבדוק כל דיסקט המוכנס למערכת המחשב **לפני** שימוש בו וכמובן שיש להשתמש בעותק חוקי מוכר בלבד.

תכניות המנקות וירוסים מציעות לתקן את הקבצים הפגועים, אך מומלץ לנקוט אמצעי זהירות מירביים לפני ההרשאה לתיקון. פעולת ניקוי

מוירוס מחייבת שינוי תוכן הקובץ, או מערכת הקבצים, ולכן היא עלולה במקרים מסוימים לפגום בהם. על כן, רק גיבוי "נקי" של הקבצים יכול להציל את התכנית "המתוקנת". אם נקפיד לנקות כל דיסקט לפני השימוש בו, נקטין את הסיכויים להידבק בוירוס.

ו. שימוש בתכניות חיסון למיניהן. תכניות אלו נועדו למנוע מתכניות הנגועות בוירוס להכנס למערכת המחשב. יש תכניות חיסון תושבות (resident), אשר נמצאות בזיכרון המחשב מרגע שנטענו לתוכו. הן מזהות וירוסים ולא מאפשרות להם לחדור למערכת. הזיהוי והחסימה מתבצעים כאשר מריצים תכנית נגועה בוירוס, או שתכנית וירוס מנסה לשמור נתונים בקבצים. יש לזכור, כי כמו כל תכנית הנמצאת בקביעות בזיכרון, גם תכניות אלו צורכות משאבי מחשב ולעתים הן עלולות להתנגש עם כלי תוכנה אחרים הקיימים במערכת.

ז. העתקות תוכנה או שימוש בתוכנה ממקור לא ידוע. הדרך הבטוחה להמנע מוירוסים היא לא להכניס למערכת המחשב דיסקטים המכילים תוכנה שאיננו יודעים מה מקורה. עותקים חוקיים של תוכנה הנרכשים מחנות לא אמורים להכיל בתוכם וירוסים ולכן מידת הסיכון בשימוש בהם מעטה.

קיימות היום בשוק תכניות רבות המסתתרות תחת שמות של תכניות ידועות וכל המשתמש בהם משתמש למעשה ב"חומר נפץ". מטרתן היא אחת: למחוק את המידע במחשב בו הן מורצות ולגרום לנזק רב ככל שניתן. תכניות שאינן מקוריות וחוקיות עושות דרכן דרך מערכות מחשבים רבות ועל כן הסיכוי שהן מכילות וירוסים גבוה במיוחד. אם הדבר אפשרי, אין לתת לאנשים נוספים להשתמש במחשב ואין להשאיל דיסקטים לשימוש במערכות אחרות!

ח. יש להקפיד על מספר כללי הגנה. אם המחשב משותף למספר משתמשים, יש להקפיד שלא לבצע BOOT מדיסקט שאיננו "מכירים", אלא רק מדיסקט שלנו המוגן בתווית הגנה ואינו מכיל וירוסים. רצוי לכבות את המחשב ולבצע BOOT לאחר שימוש של אחרים בו, כי תהליך זה מוחק כל וירוס אם נמצא בזיכרון המחשב משימוש קודם.

7.6 תכניות לטיפול בוירוסים

בשוק התוכנות לטיפול בוירוסים קיימות תכניות אנטי-וירוס רבות, אשר חדשות מצטרפות אליהן מדי פעם. תכניות אנטי-וירוס אטרקטיביות למשתמשים בשל עלות ההשקעה הגבוהה בתוכנות ובנתונים האגורים במחשב ומידת הנזק הגדולה שעלולים לגרום הוירוסים. לא כל תכניות האנטי-וירוס יודעות לאתר את כל הוירוסים הקיימים, ולא כל תכנית היודעת לאתר וירוס, יודעת גם לנטרל אותו בצורה טובה, או לשחזר את מצב הקובץ הפגוע.

תמיד יהיו וירוסים חדשים, אשר אינם מוכרים לתכנית אנטי-וירוס ותמיד יוצעו בשוק תכניות אנטי-וירוס חדשות. יש להימנע משימוש בתכניות לא מוכרות, אשר עלולות לגרום נזקים למערכת המידע במחשב. מומלץ להשתמש רק בעותק חוקי של תכנית אנטי-וירוס, ולא להשתמש בתכניות מועתקות, או מפוצחות. התועלת שבשימוש בתכנית אנטי-וירוס מקורית מצדיקה את ההשקעה, לעומת הנזק האפשרי הצפוי.

קורים מקרים בהם תיקון של קובץ ומחיקת הוירוס ממנו גורמים לו נזק ולכן מומלץ להשתמש בתכניות אנטי-וירוס בזהירות ולדאוג תמיד לגיבוי נאות של התוכנה והנתונים. חלק מתכניות החיסון עלולות למנוע מתכניות חוקיות להתבצע, מכיון שהן משתמשות במנגנון הדומה לזה של וירוסים כאשר הן נמצאות דרך קבע בזיכרון. מספר תכניות חיסון מתריעות על ניסיונות טעינה של תכנית תושבת ומאפשרות למשתמש להחליט על דרך הטיפול.

מומלץ להכניס את הבדיקה של הדיסק בעזרת תכנית האנטי-וירוס לקובץ AUTOEXEC.BAT כדי לבדוק באופן אוטומטי את מערכת המחשב ולנקותה מוירוסים, אם נמצאו, בכל פעם שמבצעים BOOT. בנוסף, יש לבדוק כל דיסקט המוכנס לכונן הדיסקטים לפני השימוש בו.

אנו חוזרים על כללי ההתנהגות: ברגע שנתגלה וירוס במערכת, או שקיים חשד, **אין להמשיך ולהשתמש במחשב.** יש לכבות מיד את המערכת, לבצע BOOT באמצעות דיסקט DOS מקורי, נקי מוירוסים ומוגן בתווית הגנה, ורק לאחר מכן לנקות את המחשב בעזרת תכנית אנטי-וירוס. הדרך הטובה ביותר להחזיר למצב עבודה תכניות שנפגעו מוירוס היא להעתיק אותן מדיסקט הגיבוי אל הדיסק הקשיח (או דיסקט), ולא להשתמש בעותק שתוקן על ידי תכנית אנטי-וירוס.

תכניות לטיפול בוירוסים ניתן לרכוש בכל חנות מחשבים. תכניות נפוצות: Anti-Virus+, ViruSafe, UnVirus, DisK Watcher, Immune, V-Analyst, Vcare ועוד.

7.7 תכנית הדגמה לוירוס

ה"תקיפה" של תכנית וירוס כוללת שלושה שלבים: שכפול, הסתתרות וחבלה. תכנית הוירוס VIRDEMO.PAS שבדוגמה משתמשת באמצעים פרימיטיבים על מנת שלא תשמש הדרכה לחבלנים. כמו כן לא הוכנסו בה אמצעי הסתתרות, כדי שאפשר יהיה לחסל אותה בקלות, מבלי לגרום לתקלות במערכת הקבצים.

שלב השכפול: כאשר התכנית נטענת לזיכרון המחשב, היא בוחנת את כל הכוננים הקיימים במערכת ומעתיקה את קובץ VIRDEMO.EXE מהספריה הנוכחית אל ספריית השורש של הכונן. שים לב, קובץ VIRDEMO.EXE הוא תכנית הוירוס עצמה, אשר **שכפלה את עצמה על ידי פעולת ההעתקה.**

שלב ההסתתרות: בדוגמה זו לא הסתרנו את הוירוס, אלא הצבנו אותו במקום גלוי. התכנית מכניסה את הפקודה VIRDEMO בתחילת קובץ AUTOEXEC.BAT של הכונן, או הכוננים שעליהם היא שכפלה את עצמה. פעולה זו גורמת לכך, שכאשר נבצע BOOT מהדיסקט/דיסק הנגוע, תכנית הוירוס תוכנס לפעולה.

שלב החבלה: <u>תכנית הדוגמה אינה הרסנית!</u> "החבלה" מתבטאת בהצגת הודעה על המסך המציינת שמערכת המחשב נגועה בוירוס וניתנת הסבר קצר כיצד להסיר אותו.

בסיום קטע זה של לימוד בעיית הוירוס, אנו מאחלים לך שהידע ישאר תיאורטי בלבד ושלא תצטרך לנצל אותו לניקוי מערכת מחשב נגועה.

אבטחת מידע
במערכות ממוחשבות

ה.פ. לוין



ספרי לימוד והכשרה במדעי המחשב
הוצאת הוד עמי

ה. פ. לוין - אבטחת מידע במערכות ממוחשבות

הוצאת הוד-עמי

פרק 8
תוכנית DEBUG

תכנית DEBUG כלולה במערכת ההפעלה DOS כתכנית חיצונית. היא משמשת למפתחי תוכנה ככלי עזר עיקרי **לניפוי שגיאות תוכנה** (Debugger).

תכנית DEBUG מאפשרת מספר פעולות המסייעות בגילוי השגיאות:

* הרצה של התכנית ברצף פקודות, או פקודה אחר פקודה.
* הצגה של האוגרים השונים במהלך ההרצה.
* הצגת קטעי זיכרון וערכי קלט/פלט מהתקני קלט/פלט במערכת.
* עדכון של קטעי זיכרון ומרחבי קלט/פלט.
* תרגום מקוד בשפת מכונה לקוד בשפת אסמבלי ולהיפך.
* טעינה של תכניות.

התכנית DEBUG שימושית לא רק בגילוי שגיאות בתכניות מוכנות. היא גם מאפשרת בדיקה של החומרה המותקנת במערכת המחשב, או אתחול של ציוד חומרה כלשהו. לדוגמה, בחלק מבקרי הדיסק הקשיח במחשבים המבוססים על מעבד 8088, תכנית האתחול (Pre-Format) של הדיסק נמצאת בבקר הדיסק. כדי לבצע אותה צריך להשתמש בתכנית DEBUG. התכנית מסוגלת לטעון ולהריץ אך ורק קבצים במבנה EXE, אך לא קבצים במבנה COM.

ההסברים שבהמשך מתייחסים לתכנית DEBUG.COM הפועלת בגירסת מערכת ההפעלה DOS 4.01.

8.1 הרצת התכנית

יש להקיש את הפקודה:

```
A>Debug
```

או

```
A>Debug File-Name.EXE
```

File-Name.EXE - שם התכנית שיש לטעון לתכנית Debug.

הקשת הפקודה תיעשה בתת-ספריה DOS, או תחת כל תת-ספריה אחרת אם הוגדרה הפנייה למסלול (Path) של תת-ספריה DOS. יש להמתין לסימן '-'. תו זה הוא הסימן המנחה של תכנית Debug. הוא מציין בכל שלב שהוא, כי התכנית מוכנה לקבל פקודות מהמשתמש.

8.2 פקודות תכנית DEBUG

בפרק זה לא נפרט את כל פקודות התכנית, אלא רק את הפקודות העיקריות והשימושיות ביותר. כך יוכל הקורא לתרגל את החומר שנלמד בפרקי הספר השונים. במהלך ההסבר, מלים וספרות **מודגשות** מציינות שנתונים אלה הוכנסו על ידי המשתמש. מלים וספרות שאינן מודגשות, מציינות נתונים הנכתבים על ידי תכנית Debug.

מושגים בתכנית:

כתובת - בכל מקום שבו מוזכרת המלה "כתובת", המשמעות היא: כתובת במרחב הכתובות, אשר מורכבת משני חלקים, **מקטע והיסט**. כלומר, היא כוללת שני מספרים כמו לדוגמה, כתובת C000:0 - אשר מציינת כתובת 0 יחסית למקטע C000Hex, או כתובת CS:100 - המציינת כתובת 100Hex יחסית לאוגר מקטע הקוד.

כתובת קלט/פלט- כתובת במרחב הקלט/פלט. זהו מספר בין הערך 000Hex לבין הערך 3FFHex.

תחום - תחום הכתובות מורכב **מכתובת בסיס**, ורצף של כתובות המציין מה אורך רצף הכתובות שאליו יש להתייחס בפקודה. קיימות שתי דרכים עיקריות להגדרת תחום (ראה דוגמאות בהמשך):

* Segment:Offset End-Offset תחום הכתובת בין הכתובת Offset לכתובת End-Offset בתוך המקטע Segment.

* Segment:Offset L Size כאשר התחום הינו בגודל Size החל מכתובת Offset היחסית למקטע Segment.

8.2.1 תקציר הפקודות

* קרא תוכן זיכרון	D **תחום**	Dump
* כתוב נתונים לזיכרון	E **כתובת**	Enter
* הצג/שנה תוכן אוגרים	R **אוגר**	Register
* קרא בית מערוץ קלט/פלט	I **כתובת-קלט/פלט**	Input
* כתוב בית לערוץ קלט/פלט	O **כתובת-קלט/פלט**	Output
* תרגם משפת מכונה לשפת אסמבלר	U **תחום**	UnAssemble
* בצע תכנית מכתובת	G=**כתובת**	Go
* עקוב אחרי ביצוע התכנית בצעדים	T=**כתובת**	Trace
* תרגם משפת אסמבלר לשפת מכונה	A **כתובת**	Assemble
* תן שם לקובץ	N **שם-קובץ**	Name
* טען קובץ	L **שם-קובץ**	Load
* מלא תוכן זיכרון בערך	F **ערכים תחום**	Fill
* צא מתכנית למערכת ההפעלה	Q	Quit

8.2.2 פקודות נבחרות

נציג את הפקודות השונות ונסביר את אופן פעולתן ואת צורת הכתיבה של הפרמטרים, והעיקר – ניתן דוגמאות. שים לב לכך שקידום כתובת קוד שפת המכונה (אוגר IP) נעשה על ידי התכנית Debug ובהתאם לאורך הפקודה.

מבנה הפקודה D תחום

D - הצגת תוכן קטע זיכרון במערכת המחשב.
תחום - תחום כתובות הצגה.

מטרת הפקודה ואופן השימוש:
הצגת תחום זיכרון ממרחב הזיכרון במחשב על גבי המסך. על המסך יוצג קטע זיכרון הכלול בתחום. הנתונים יוצגו ב"הקסה" ובהצגת ASCII (תווים). בתחילת כל שורה תוצג כתובת הבסיס של אותה שורה במרחב הזיכרון במחשב.

דוגמאות:

```
-D B000:0 L 50
B000:0000  20 07 20 07 20 07 20 07-20 07 20 07 20 07 20 07
B000:0010  20 07 20 07 20 07 20 07-20 07 20 07 20 07 20 07
B000:0020  20 07 20 07 20 07 20 07-20 07 20 07 20 07 20 07
B000:0030  20 07 20 07 20 07 20 07-20 07 20 07 20 07 20 07
B000:0040  20 07 20 07 20 07 20 07-20 07 20 07 20 07 20 07
-
```

הפקודה תציג על גבי המסך 50Hex בתים ממרחב הזיכרון החל בכתובת B0010Hex ועד כתובת B005FHex. מכיון שתוכן כרטיס מסך מונוכרום נמצא בכתובת B000:0, יוצגו על המסך 37 תווים ראשונים וליד כל תו יוצג ערך המציין את תכונותיו (Attributes) כפי שהוא מוצג במסך.

פקודה:

```
-D 40:0 7
0040:0000  F8 03 00 00 00 00 00 00
-
```

פקודה זו תציג על גבי המסך את הכתובות של 4 כרטיסי התקשורת הטוריים הקיימים במחשב (Com1 עד Com4). הפלט המוצג בדוגמה מתייחס למחשב שקיים בו כרטיס טורי יחיד, שכתובתו היא 03F8Hex.

פקודה:

```
-D 40:8 D
0040:0000  BC 03 00 00 00 00
-
```

הפקודה תציג על גבי המסך את הכתובות של 4 כרטיסי המדפסת המקבילים הקיימים במחשב (LPT1 עד LPT3). במחשב שבדוגמה המתוארת קיים כרטיס מקבילי יחיד בכתובת: BC03Hex.

פקודה:

```
-D 40:13 14
0040:0010  7F 02
-
```

הפקודה תציג על המסך את גודל הזיכרון הבסיסי המותקן במחשב. במחשב שבדוגמה קיים זיכרון בסיסי של 639KByte (הערך העשרוני 027FHex=639).

פקודה:
```
-D 40:17 17
0040:0010  E0
-
```

הפקודה תציג על המסך את מצב הדגלים השונים של המקלדת (Alt ,Ins וכו'). על פי הדוגמה, ערך המספר הוא E0 ולכן 3 הסיביות השמאליות שלו הן 1. לכן, הדגלים מסמנים שמקש Insert במצב ON, מקש Caps Lock במצב ON ומקש Num Lock במצב ON.

פקודה:
```
-D 40:49 49
0040:0040  07
-
```

הפקודה תציג על גבי המסך את אופן הפעולה של המסך (Crt Mode). הבדיקה בוצעה על מחשב עם כרטיס מתאם מסך מונוכרום ולכן צורת הפעולה של המסך מוגדרת 7.

פקודה:
```
-D 40:6C 6F
0040:0060  D2 6A 11 00
-
```

הפקודה תציג על גבי המסך את ערך השעון הפנימי במחשב (שעון התוכנה). כדי לקבל את השעה יש לבצע את הפעולה בהקסה 00116AD2/FFF0. בדוגמה זו נקבל 11Hex שהם 17, כלומר השעה 5 אחה"צ.
כדי לקבל את הדקות יש לחלק את השארית ב-FFF0 בערך 0444.

בדוגמה (00116AD2 MODULO FFF0)/444 = 25
כלומר, השעה 17:25.

<u>הסבר:</u> פסיקת שעון התוכנה מתרחשת 18.2 פעמים בשנייה. יש 60 שניות בדקה ולכן נקבל 1092 פסיקות בדקה (444Hex פסיקות). בשעה יש 60 דקות, ולכן נקבל 65520 פסיקות בשעה (0FFF0Hex פסיקות).
ראה תכנית הדוגמה CLOCK.

מבנה הפקודה

E כתובת

או

E ערך כתובת

E - הכנסת ערכים ושינוי תוכן הזיכרון.
כתובת - כתובת התחלתית בזיכרון.
ערך - ערך שיש להכניס לכתובת.

מטרת הפקודה ואופן השימוש:
על המסך תופיע כתובת בזיכרון ובצידה יופיע ערך הנתון הקיים בכתובת זו (נתון ברוחב 8 סיביות). מספר אפשרויות קיימות במצב זה:

* נזין נתון ולאחריו נקיש Enter: הנתון יכנס לזיכרון בהתאם לכתובת. תהליך עדכון תוכן הזיכרון ייפסק ונקבל את הסימן המנחה של DEBUG.

* הכנסת נתון והקשה על מקש רווח בסיום הכנסתו: הנתון יעודכן בזיכרון ותוצג הכתובת הבאה להכנסת נתון. ניתן להמשיך ברצף באותו האופן לעדכון של קטעים נוספים. כדי לסיים את הכנסת נתונים יש להקיש על מקש Enter.

* הקשה על מקש רווח ללא הכנסה של נתון תגרום לכך שתוכן הזיכרון בכתובת המתאימה לא ישתנה ונמשיך ברצף לכתובת הבאה. הקשת Enter ללא הכנסת נתון תגרום לכך שתוכן הכתובת המתאימה לא ישתנה ותהליך העדכון ייפסק.

לדוגמה נקיש את הפקודה הבאה:

```
                   הקשה על
-E B07D:0          מקש רווח                  מקש ENTER
                       |                           |
B07D:0000   20.41   07.70   20.42   07.70   20.
               |       |       |       |
-             התו   תכונת     התו   תכונת
               A       A       B       B
```

הפקודה תאפשר הכנסת נתונים ברוחב של 8 סיביות (בית) החל מכתובת B07D. הקשת נתון ולחיצה על Enter תכניס אותו לכתובת זו ותהליך העדכון יסתיים. הקשת נתון ולחיצה על מקש רווח תכניס אותו לכתובת ותאפשר להמשיך בתהליך העדכון.

מכיון שכתובת B07D:0 היא כתובת התו של אמצע המסך במתאם מסך MONO, ערך הנתון ראשון שנכניס ישנה את התו המוצג בפינה השמאלית העליונה של המסך. הערך השני שנכניס ישנה את צורת ההצגה של אותו התו (תכונה) על גבי המסך. אם קיים ברשותך מתאם מסך צבע, יש לרשום B87D:0 במקום B07D:0. בדוגמה יוצגו התווים AB ברצף ברקע לבן (Reverse).

מבנה הפקודה R

או

R Register-Name

R - הפקודה תאפשר להציג את תוכן האוגרים של המעבד. (ראה פרק על מבנה מעבד).

Register-Name - שם האוגר שיש להציגו או לשנות את תוכנו.

מטרת הפקודה ואופן השימוש:
הפקודה מאפשרת להציג את תוכן האוגרים השונים של המעבד לאחר ביצוע פעולה כלשהי. מעקב זה מאפשר לבדוק את נכונות הביצוע של תכנית בשפת מכונה ולהציג את דגלי המעבד בעקבות ביצוע הפקודה. כמו כן, ניתן לשנות תוכן אוגר תוך כדי ביצוע תכנית. אם לא מוזכר שם האוגר, יוצגו כל האוגרים. שמות אוגרים האפשריים הם: AX, BX, CX, DX, CS, ES, DS, SS, BP, SP, SI, DI, F, PC.

דוגמאות:

הצגת כל האוגרים והדגלים

```
-R
AX=0000 BX=0000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000  DI=000
DS=3293 ES=3293 SS=3293 CS=3293 IP=0100 NV UP EI PL NZ NA PO NC
```

הצגה ושינוי ערך אוגר AX

```
-R  AX
AX 0000
:1234
```

```
-R
AX=1234 BX=0000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000  DI=000
DS=3293 ES=3293 SS=3293 CS=3293 IP=0100 NV UP EI PL NZ NA PO NC
```

הצגה ושינוי ערך אוגר BX

```
-R  BX
BX  0000
:4321
-R  BX
BX  4321
:
-
```

כאשר מבקשים להציג תוכן אוגר יחיד מוצג תוכנו ולאחריו הסימן ":". בשלב זה ניתן להכניס ערך כדי לשנות את תוכן האוגר. בדוגמה, לאוגר AX הכנסנו את הערך 1234Hex ולאוגר BX הכנסנו את הערך 4321Hex.

מבנה הפקודה:

U תחום

או

U

או

U כתובת

U - תרגום מקוד בשפת מכונה לקוד בשפת אסמבלי (קיצור UnAssemble).
תחום - תחום הכתובות שיש לתרגם.
כתובת - כתובת התחלתית בתחום אותו יש לתרגם.

מטרת הפקודה ואופן השימוש:
הפקודה נועדה לתרגם תוכן קטע קוד בזיכרון משפת מכונה לשפת אסמבלי, כדי שההצגה התכנית לבדיקה תהיה בשפה מובנת יותר למתכנת. התכנית Debug תתרגם את תחום הכתובות הנדרש לשפת אסמבלי ותציג את הפקודות זו אחר זו. כל פקודה תרשם בשורה נפרדת בלווית הכתובת שבה היא נמצאת. אם לא נרשום כתובת או תחום, תתרגם התכנית את הקוד החל מהמקום שעליו מצביע אוגר IP, יחסית למקטע הקוד (CS).

דוגמה: הפקודה הבאה מציגה את תרגום קוד המכונה של הקובץ SAMPLE.EXE לשפת אסמבלי.

הרץ את התכנית

```
A>DEBUG  SAMPLE.EXE
-U

32E0:0000 1E          PUSH DS
32E0:0001 2BC0        SUB  AX,AX
32E0:0003 50          PUSH AX
32E0:0004 B8BF32      MOV  AX,32BF
32E0:0007 8ED8        MOV  DS,AX
32E0:0009 B300        MOV  BL,00
32E0:000B 8D360000    LEA  SI,[0000]
32E0:000F B91000      MOV  CX,0010
32E0:0012 B400        MOV  AH,00
32E0:0014 CD16        INT  16
32E0:0016 FEC3        INC  BL
32E0:0018 3C0D        CMP  AL,0D
32E0:001A 7405        JZ   0021
32E0:001C 8804        MOV  [SI],AL
32E0:001E 46          INC  SI
32E0:001F E2F1        LOOP 0012
-
```

הערכים בתכנית DEBUG מוצגים תמיד בקוד הקסה-דצימלי.

מבנה הפקודה I כתובת-קלט/פלט

I - פקודה לקריאת נתון ממרחב הקלט/פלט.
כתובת-קלט/פלט - כתובת מרחב הקלט/פלט, בהקסה.

מטרת הפקודה ואופן השימוש:
קריאת נתון ממרחב הקלט/פלט בכתובת כלשהי נתונה. על גבי המסך יוצג נתון ברוחב של 8 סיביות (בית) שנקרא החל מכתובת "כתובת-קלט/פלט" שבמרחב הקלט/פלט.

דוגמה:

```
-I 21
B8
-
```

הפקודה תציג על גבי המסך נתון שקראה מהתקן קלט המקושר לכתובת 21Hex שבמרחב כתובות קלט/פלט. בכתובת 21Hex מקושר בקר הפסיקות הראשון המספר B8Hex מציין אילו פסיקות חומרה מאופשרות (Enabled) בבקר. בדוגמה מאופשרות פסיקות IRQ0 המשמשת את שעון הזמן, פסיקה IRQ1 המשמשת את לוח המקשים ופסיקה IRQ2 שמשרשרת את פסיקות בקר הפסיקות השני (התכנית בוצעה במחשב AT).

מבנה הפקודה   O כתובת-קלט/פלט Value

O - פקודת לכתיבת נתון להתקן פלט המחובר למרחב קלט/פלט.
כתובת-קלט/פלט - כתובת בהקסה של ההתקן במרחב הקלט/פלט.
Value - ערך בגודל של בית שיש לרשום החל מ"כתובת-קלט/פלט" שבמרחב הקלט/פלט.

מטרת הפקודה ואופן השימוש:
כתיבת נתון למרחב הקלט/פלט בכתובת הרצויה. להתקן במרחב הקלט/פלט יכתב נתון ברוחב של 8 סיביות (בית).

דוגמה:
הפקודות הבאות ידפיסו את התו A על גבי המדפסת המקבילית. בגמר כתיבת התו A נשלח למדפסת הצירוף CR+LF, כדי לגרום למדפסת להדפיס את התו ולרדת שורה בדף. הכתיבה נעשית ישירות לכתובת הכרטיס המתאם המקבילי.

מציאת כתובת הכרטיס המתאם למדפסת	-D 40:8 9 0040:0000 BC 03
התו A	-O 3BC 41
בקרה למדפסת	-O 3BE D
בקרה למדפסת	-O 3BE C
התו CR	-O 3BC D
בקרה למדפסת	-O 3BE D
בקרה למדפסת	-O 3BE C
התו LF	-O 3BC A
בקרה למדפסת	-O 3BE D
בקרה למדפסת	-O 3BE C
	-

מבנה הפקודה   G=כתובת

או

G

או

G כתובת

G - בצע את קוד שפת המכונה.
כתובת - כתובת התחלתית לביצוע הקוד. ברירת המחדל היא CS:IP.

מטרת הפקודה ואופן השימוש:
הפקודה מאפשרת לבצע הרצה של תכנית מכתובת מסוימת, או עד כתובת מסוימת. פקודה זאת שימושית כשרוצים לבדוק נכונות של קטעי תכנית או כשרוצים להריץ קטע תכנית שאין כל דרך אחרת להגיע אליו. דוגמה להרצה כזו יכולה להיות הרצת תכנית המבצעת Pre-Format לדיסק קשיח ונמצאת בבקר הדיסק, ולא בתוך קובץ שפת מכונה.

דוגמאות:

הרץ את התכנית

```
A>DEBUG SAMPLE.EXE
-u

32E1:0000 1E          PUSH DS
32E1:0001 2BC0        SUB  AX,AX
32E1:0003 50          PUSH AX
32E1:0004 B8C032      MOV  AX,32C0
32E1:0007 8ED8        MOV  DS,AX
32E1:0009 B300        MOV  BL,00
32E1:000B 8D360000    LEA  SI,[0000]
32E1:000F B91000      MOV  CX,0010
32E1:0012 B400        MOV  AH,00
32E1:0014 CD16        INT  16
32E1:0016 FEC3        INC  BL
32E1:0018 3C0D        CMP  AL,0D
32E1:001A 7405        JZ   0021
32E1:001C 8804        MOV  [SI],AL
32E1:001E 46          INC  SI
32E1:001F E2F1        LOOP 0012
```

הפקודה בכתובת 9

קרא מקש מלוח מקשים

הפקודה בכתובת 1E

```
-g 9
```

בצע עד כתובת 9 יחסית למקטע הקוד

```
AX=32C0 BX=0000 CX=023D DX=0000 SP=01FC BP=0000 SI=0000  DI=000
DS=32C0 ES=32B0 SS=32C1 CS=32E1 IP=0009 NV UP EI PL ZR NA PE NC

32E1:0009 B300        MOV  BL,00
-g 1e
```

בצע עד כתובת 1E יחסית למקטע הקוד
כאן התכנית מחכה שנקיש על מקש כלשהו.
נקיש על מקש A שערכו 41Hex לפי טבלת ASCII, ונקבל:

```
AX=1E41 BX=0001 CX=0010 DX=0000 SP=01FC BP=0000 SI=0000  DI=000
DS=32C0 ES=32B0 SS=32C1 CS=32E1 IP=001E NV UP EI PL NZ AC PO NC

32E1:001E 46          INC  SI

-D DS:0 F
```

הצגת תוכן מקטע הנתונים

```
32C0:0000 41 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
-
```

ערכו של מקש A מוצג במקטע הנתונים (41Hex)

מבנה הפקודה

T

או

T=כתובת

או

T=ערך כתובת

או

T ערך

T . - עקוב אחר התכנית ובצע פקודה אחת, או מספר פקודות מוגדר מראש. (קיצור של המלה Trace).
כתובת - כתובת התחלה של קוד התכנית שיש לבצע.
ערך - מספר הפקודות שיש לבצע. ברירת המחדל: פקודה אחת.

מטרת הפקודה ואופן השימוש:
ביצוע של מספר פקודות מוגדר מראש. עבור כל ביצוע של פקודה, הצגת תוכן כל האוגרים במעבד ועצירה לאחר הביצוע. הפקודה T מאפשרת לבצע את התכנית צעד אחר צעד ולבדוק בכל שלב את צורת הביצוע ותוצאתו ובאופן זה לבדוק אם התכנית מתבצעת כראוי.

דוגמאות:

טעינת תכנית הדוגמה

```
A>DEBUG SAMPLE.EXE
```

ביצוע פקודה אחת

```
-T
AX=0000 BX=0000 CX=023D DX=0000 SP=01FE BP=0000 SI=0000  DI=000
DS=32B0 ES=32B0 SS=32C1 CS=32E1 IP=0001 NV UP EI PL NZ NA PO NC

32E1:0001 2BC0          SUB  AX,AX
```

ביצוע רצוף של שתי פקודות

```
-T2
AX=0000 BX=0000 CX=023D DX=0000 SP=01FE BP=0000 SI=0000  DI=000
DS=32B0 ES=32B0 SS=32C1 CS=32E1 IP=0003 NV UP EI PL ZR NA PE NC

32E1:0003 50            PUSH AX

AX=0000 BX=0000 CX=023D DX=0000 SP=01FC BP=0000 SI=0000  DI=000
DS=32B0 ES=32B0 SS=32C1 CS=32E1 IP=0004 NV UP EI PL ZR NA PE NC

32E1:0004 B8C032        MOV  AX,32C0
-
```

מבנה הפקודה: A כתובת

A - תרגם קוד בשפת אסמבלר לקוד בשפת מכונה (קיצור של Assembly).
כתובת - כתובת התחלתית בזיכרון. מכתובת זאת יש להתחיל לרשום את הקוד שפת המכונה.

מטרת הפקודה ואופן השימוש:
כתיבת תכנית, או פקודות בשפת אסמבלי ותרגומן המיידי לשפת מכונה של המחשב כדי שניתן יהיה לבצען. כתיבה בשפת אסמבלי פשוטה יותר עבור המשתמש מאשר כתיבה בשפת מכונה. לאחר הקשת הפקודה A, ניתן לרשום פקודות באסמבלי. התכנית Debug תתרגם אותן לשפת מכונה ותקדם את מצביע הכתובת (אוגר IP) על פי אורך הפקודה שהוכנסה.

על מנת להבדיל בין סוגי הפקודות השונים של המעבד, יש להוסיף קידומת של Near ו-Far לפני סיעוף (GOTO) וקידומת של Word ו-Byte במיעון

לזיכרון. יש להקיף כתובת בסימנים [כתובת] על מנת להבדילה מערך קבוע (מיעון עקיף מושג על ידי שימוש במיקוף).

דוגמה:
אוסף הפקודות הבא יאפשר ליצור תכנית בשפת מכונה. הפקודות מוכנסות בשפת אסמבלי ותכנית Debug מתרגמת אותן לשפת מכונה. התכנית מיועדת לבצע **חניה** (PARK) לדיסק הקשיח. "חניית הדיסק" הינה הזזה של מערכת הראשים הקוראים של הדיסק הקשיח לצילינדר האחרון ונעילה שלהם. ללא פעולה זו, עלול הדיסק להינזק כאשר מטלטלים את המחשב ממקום למקום.

הכנסת פקודות בשפת אסמבלי

```
-A
3293:0100  MOV AH,19
3293:0102  MOV DL,80
3293:0104  INT 13
3293:0106  NOP
3293:0107
```

פסיקת שירות לנעילת ראשים (ליד השורה INT 13)

הצגת התכנית

```
-U
3293:0100 B419          MOV  AH,19
3293:0102 B280          MOV  DL,80
3293:0104 CD13          INT  13
3293:0106 90            NOP
3293:0107 50            PUSH AX
3293:0108 8CC0          MOV  AX,ES
3293:010A 051000        ADD  AX,0010
3293:010D 8B0E1E01      MOV  CX,[011E]
3293:0111 03C8          ADD  CX,AX
3293:0113 894FFB        MOV  [BX-05],CX
3293:0116 8B0E2601      MOV  CX,[0126]
3293:011A 03C8          ADD  CX,AX
3293:011C 894FF7        MOV  [BX-09],CX
3293:011F 8B0E2001      MOV  CX,[0120]
```

כתובת 106 (ליד השורה 3293:0106)

בצע עד כתובת 106

```
-G106

AX=0000 BX=0000 CX=0000 DX=0080 SP=FFEE BP=0000 SI=0000  DI=000
DS=3293 ES=3293 SS=3293 CS=3293 IP=0106 NV UP EI PL ZR AC PE NC

3293:0106 90            NOP
```

פקודה אחרונה בתכנית

בשלב זה ראשי הדיסק נעולים
וניתן לכבות את המחשב ולהזיז אותו למקום אחר.

מבנה הפקודה N שם-קובץ

N - מתן שם לקובץ התייחסות בתכנית Debug. לאחר פקודה זו, פקודת Load תתייחס לשם הקובץ שהגדרנו.

מטרת הפקודה ואופן השימוש:
הפקודה מאפשרת לתת שם לתכנית שאליה מתייחסים בתכנית Debug. שם קובץ נדרש כאשר רוצים לטעון תכנית בשפת מכונה לסביבת העבודה של תכנית Debug.

<u>הערה:</u>
צריך לתת שם קובץ, אם הפעלנו את תכנית Debug בלא להזכיר את השם של תכנית בשפת מכונה שאותה יש לטעון לשם בדיקה. בלעדי פקודה זו לא נוכל לטעון תכנית לתוך סביבת העבודה של Debug.

מבנה הפקודה L

L - טען קובץ שפת מכונה בסיומת EXE לתכנית Debug.

מטרת הפקודה ואופן השימוש:
פקודה זו מאפשרת לטעון תכנית הכתובה בשפת מכונה לסביבת העבודה של תכנית Debug על מנת שאפשר יהיה לבדוק ולבצע אותה. שם הקובץ ניתן על ידי פקודה N.

לדוגמה, נכתוב את רצף הפקודות:

```
-N  SAMPLE.EXE
-L
-
```

פקודות אלו תאפשרנה לטעון את הקובץ SAMPLE.EXE הכתוב בשפת מכונה לתוך תכנית Debug.

מבנה הפקודה F תחום סדרת-ערכים

F - מילוי תחום כתובות בערך, או בסדרת ערכים נתונה.
תחום - תחום הכתובות שאותו יש למלא בערך, או בסדרת הערכים.
סדרת-ערכים - הערך או סדרת הערכים שיש לכתוב בתחום הכתובות בזיכרון.

מטרת הפקודה ואופן השימוש:
פקודה זו מאפשרת למלא תוכן של זיכרון בערך כלשהו רצוי, או בסדרת ערכים. אם נכתוב ערך בודד - כל התחום יתמלא באותו הערך, ואם נכתוב סדרת ערכים - הסדרה תחזור על עצמה לאורך כל תחום הכתובות הנזכר בפקודה.

דוגמאות:
הפקודות הבאות יטענו את תכנית SAMPLE.EXE, יריצו אותה עד להגדרת מקטע הנתונים וימלאו את הזיכרון בעזרת הפקודה F. בשלב ראשון ימולא מקטע הנתונים (10Hex בתים) בערך FF ובשלב השני – בסדרת המספרים: 10 20 30 40.

טעינת תכנית הדוגמה

```
A>DEBUG SAMPLE.EXE
```

הרצה עד לפקודה המגדירה את מקטע הנתונים על מנת שלא נפגע בתוכן זיכרון החיוני ל-DOS.

```
-G9

AX=32C0 BX=0000 CX=023D DX=0000 SP=01FC BP=0000 SI=0000 DI=0000
DS=32C0 ES=32B0 SS=32C1 CS=32E1 IP=0009 NV UP EI PL ZR NA PE NC
32E1:0009 B300          MOV  BL,00
```

הצגת מקטע הנתונים

```
-d ds:0 f
32C0:0000  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
```

מילוי מקטע הנתונים בערך FF

```
-f ds:0 10 ff
```

הצגת מקטע הנתונים

```
-d ds:0 f

32C0:0000  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
```

מילוי מקטע הנתונים בסדרת מספרים

```
-f ds:0 1 10 10 20 30 40
```

הצגת מקטע הנתונים

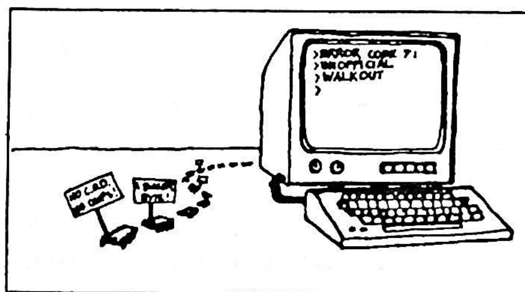
```
-d ds:0 f

32C0:0000  10 20 30 40 10 20 30 40-10 20 30 40 10 20 30 40
-
```

מבנה הפקודה Q

Q - יציאה מתכנית Debug וחזרה למערכת ההפעלה DOS.

מטרת הפקודה ואופן השימוש:
יציאה מתהליכי הניפוי של תכניות המשתמש וחזרה למערכת ההפעלה.



הודעה: ה"ג'וקים" נוטשים!

פרק 9
תקשורת במחשבים אישיים

המחשב האישי מוגדר כמחשב **למשתמש יחיד ומשימה יחידה** (Single User & Single Task). כאשר הוא פועל כמחשב יחיד וממלא פונקציה או פונקציות, אשר אינן תלויות בנתונים הנמצאים במחשבים אחרים, הוא ממלא את יעודו בצורה הטובה ביותר.

בתחילה לא נדרשו כל משימות נוספות. ככל שהלך והתרחב השימוש במחשבים האישים התעורר צורך לחבר בין המחשבים לבין עצמם, לשתף מדפסות, דיסקים ונתונים בין המחשבים, לחבר את המחשבים האישיים כמסופים למחשבים מרכזיים ועוד. צרכים חדשים אלה יצרו פתרונות שונים ומגוונים שנועדו לפתור בעיות שונות ולענות על צורכיהם של המשתמשים השונים.

דרך החיבור של המחשב האישי למערכות מחשבים אחרות תלויה בצורכי המשתמש, בצורת העבודה הנדרשת ובתצורת החומרה הקיימת במחשבי הארגון.

9.1 לאן יכול המחשב האישי להתקשר?

מחשבים אישיים ניתן לקשור למחשבים אחרים בדרכים שונות:

1. רשת תקשורת מקומית – LAN (Local Area Network).
2. רשת תקשורת רחבה – WAN (Wide Area Network).
3. חיבור ישיר אל מחשב מרכזי.

בהמשך נדון בכל אחד מהנושאים בנפרד, אך עתה נציין אפיון כללי לסוגי הקשר השונים. ההבדל בין LAN לבין WAN מתבטא בטכנולוגיה המשמשת להעברת הנתונים. הטכנולוגיה אשר מיושמת ב-LAN תומכת במהירות העברה גבוהה על חשבון מרחק קצר. מהירויות שידור ברשת LAN יכולה להגיע עד 300MBS (MBS – מיליון סיביות לשניה, שהם כ-125,000 בתים) למרחק של קילומטרים אחדים. השידור ברשת מסוג WAN נעשה בקצב נמוך מ-500KBS (KBS – אלף סיביות לשניה, שהם כ-125 בתים לשניה), אבל הוא יכול להגיע לכל מקום על פני כדור הארץ.

ההתפתחות הטכנולוגית המהירה מאפשרת להגדיל את טווח ה-LAN ולהגדיל את קצב ה-WAN. בפועל, שינויים של הגדלת הטווח והגברת הקצב נעשים כל הזמן. תקן **"רשת דיגיטלית לשירותים מתושלבים"** – ISDN (Integrated Services Digital Network), שהוא תקן העתיד לתקשורת משרדית, משלב במידה מסוימת בין LAN לבין WAN.

9.1.1 שער גישור – GATEWAY

שער הגישור (Gateway) בתקשורת הוא התקן המחבר בין שתי מערכות תקשורת נפרדות. בפועל הוא מקשר בין מחשבים במערכת תקשורת אחת לבין מחשבים במערכת תקשורת אחרת. שער גישור אידיאלי יוצר אצל כל אחד מהמשתמשים את ההרגשה, שכולם פועלים באותה רשת תקשורת.

ישנם מספר סוגים של שערי גישור: בין רשתות LAN לבין רשתות ניהול מסופים במחשבים מרכזים, בין רשתות LAN לבין WAN, בין LAN לבין LAN ועוד. שערי גישור אשר משמשים לקישור בין מערכות מחשבים רחוקות איפשרו למשל, לנערים שישבו בביתם בחוף המערבי בארה"ב, לחדור למחשב בפנטגון שנמצא בחוף המזרחי.

9.2 העברת הנתונים בערוץ התקשורת

שידור הנתונים בקו התקשורת מאופיין על ידי סוגי האותות החשמליים העוברים בו (זרמים, מתחים, תדרים וכו'). שתי שיטות עיקריות מאפיינות שידור:

1. פס-בסיס (Base Band)

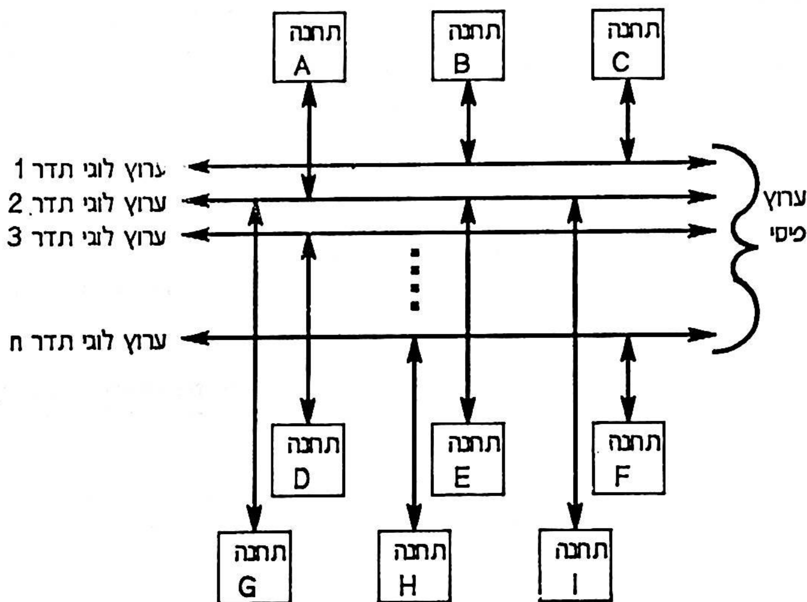
בשיטת פס-בסיס קיים ערוץ תקשורת פיסי יחיד המקשר בין המערכות. הוא מאפשר העברה של מספר ערוצי תקשורת לוגיים באמצעות חלוקה, או הקצאה, של זמן השידור ביניהם. כל הערוצים הלוגיים מאפשרים התקשרות, אך בזמנים שונים, בשיטת **שיתוף זמן** (Time Sharing).

שיתוף זמן בשיטת פס-בסיס נעשית על ידי חלוקה של ציר הזמן לחריצים שונים. כל ערוץ לוגי מקבל **חריץ זמן** (Time Slot) אחד, או כמה חריצי זמן, שבהם הוא יכול להעביר את הנתונים שלו. בדרך זו יכולים כל ההתקנים המחוברים לערוצים הלוגיים להתקשר באמצעות ערוץ התקשורת הפיסי, אך לא בו-זמנית. כמות הנתונים המירבית שניתן להעביר דרך ערוץ לוגי בזמן נתון תלויה בכמות הערוצים הלוגיים ובכמות הנתונים המירבית שיכול הערוץ הפיסי להעביר. זוהי **התפוקה** (Throughput) של הערוץ.

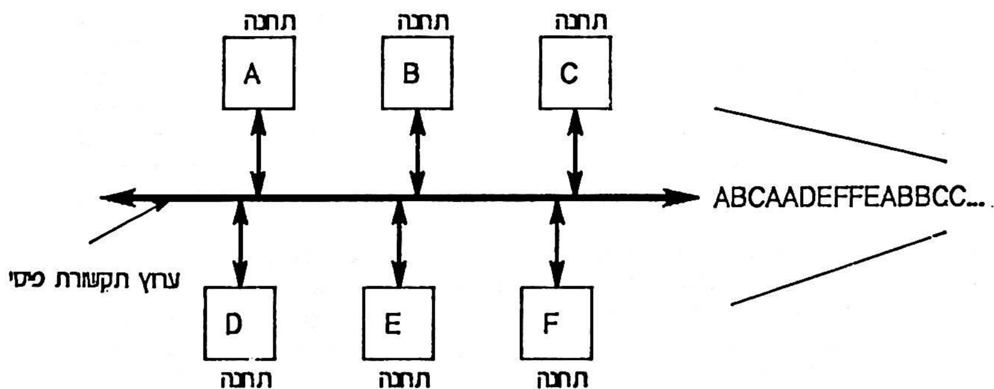
בשיטת פס-בסיס אין אפנון בין הנתונים לבין תדר של גל נושא כלשהו. הדוגמה הנפוצה היא תקן RS232 שהוסבר בפרק ג'. במספר מערכות LAN שפועלות בפס בסיס נמצא קידוד של הנתונים יחד עם תדר השעון. זהו אומנם אפנון (Modulation), אבל לא עם גל נושא.

2. פס-רחב (Broad Band)

בשיטת פס-רחב קיים ערוץ תקשורת פיסי יחיד המאפשר העברה של מספר ערוצי תקשורת **בו-זמנית**. העברה זו מושגת על ידי הקצאה של תדרים שונים לשידור על ידי הערוצים השונים. כך משיגים העברת כמות נתונים גבוהה יותר מאשר בערוץ תקשורת הפועל בשיטת פס-בסיס. מערכות פס-רחב מיועדות לרוב לשידור למרחקים גדולים.



ערוץ תקשורת בשיטת פס בסיס – BaseBand



ערוץ תקשורת בשיטת פס רחב – Broadband

בשיטת פס-רחב מאפננים את הנתונים בעזרת תדר גל נושא. ניתן ליישם העברת נתונים בשיטת פס-בסיס על ערוץ בפס-רחב, אך לא ניתן ליישם תקשורת פס-רחב על פס-בסיס. הכלי המאפשר ליישם פס-בסיס על פס-רחב נקרא **מודם** (MODEM).

דוגמה למערכת פס-רחב היא מערכת העברת נתונים במערכות הטלוויזיה בכבלים המוכנסות עכשיו לשימוש בארץ. במערכות כאלו, הפועלות בשיטת הפס-הרחב כל משתמש במחשב לקבל תדר נפרד. בעזרת מודם מיוחד המחשב הזה יכול להתחבר לכל נקודה ברשת הכבלים, מבלי להפריע לשידורי הטלוויזיה ולשידורי המחשבים האחרים באותה רשת.

9.3 תקנים לרשתות תקשורת ומודל 7 השכבות של OSI

בתקשורת קיימת משמעות מרכזית לתקנים, אשר מאפשרים לפריטי ציוד ותוכנה שונים של אותו יצרן, או של יצרנים שונים, ל"דבר" זה עם זה בנוסח "עם קצת רצון ותקן תקשורת יכול גם פיל לדבר עם תרנגולת". כאשר מוגדר תקן תקשורת, יכול כל יצרן ליצר חבילת חומרה ותוכנה המתאימה לתקן זה והצרכנים יכולים לרכוש ולהתקין מוצרים של יצרנים שונים ללא חשש של אי תאימות. כאשר פועלים לפי תקן מוסכם, יכול משתמש במחשב מסוג א' לדבר עם משתמש במחשב מסוג ב' ובלעדי תקן כזה לפנינו מגדל בבל של תקשורת.

הבעיה בתקינה נובעת מכך שמסיבות של תחרות כלכלית היצרנים יוצאים בדרך כלל לשוק עם מוצרים לפני השלמת התקינה. כך הדבר גם ברשתות תקשורת. אבל, מכיון ש**בתקשורת התקינה הכרחית** לביצוע המשימה, היצרנים משלימים בסופו של דבר עם דרישות התקינה ומתאימים את עצמם להנחיותיה. הדוגמה הטובה ביותר היא הכרזה של חברת יבמ בסוף 1990 על Gateway בין רשת Token Ring שלה לבין רשתות בתקן IEEE 802.3.

המסגרת הפורמלית לתקשורת בין מחשבים ידועה כ"**מודל 7 השכבות של OSI**". OSI היא תת-ועדה של CCITT, שהינה ועדת התקינה האירופאית העליונה לנושאי אלקטרוניקה. מערכות התקשורת של כל החברות פועלות כיום במסגרת מודל השכבות של OSI. ישנן מערכות תקשורת ששינו במקצת את ישום השכבות במודל, אבל המבנה הכללי שלו נשמר. אנו מפנים את תשומת לב הקורא לספרות הענפה הקיימת בנושאים אלה.

ניקח לדוגמה קישור בין המחשב שבתוך הכספומט לבין המחשב המרכזי של הבנק. המשתמש רואה לפניו את השיחה המתקיימת, שבה הכספומט שואל את המחשב המרכזי אם אפשר לתת לו 100 שקל, והמחשב המרכזי עונה ב"כן" או "לא". זהו קשר ברמת היישום.

ברמה הפיסית, או קו התקשורת המחבר בין שני המחשבים, עוברים תדרים שונים המיוצרים על ידי המודמים שבשני הקצוות. המעבר בין רמת היישום של המשתמש בקצה אחד לבין המשתמש בקצה השני (במקרה זה – תכנית מחשב) מחייב מספר פעולות ניהול ובקרה, אשר נעשות בשכבות שבין רמת היישום לבין הרמה הפיסית. תהליכים אלה מוגדרים במודל השכבות של OSI.

השכבות של OSI מגדירות תקשורת עקרונית כלשהי, ולכן יש צורך להתאים את התקן לכל סוג תקשורת. לרשת WAN הפועלת על פי OSI נקבע תקן X25. לרשת

LAN הפועלת לפי תקן OSI נקבע תקן IEEE 802, שהוא היום התקן החשוב ביותר ברשתות אלו. נתבונן כיצב התקן מגדיר את מבנה ארבעת השכבות התחתונות של התקשורת ברשתות LAN.

מס'	שכבת OSI		ישום על ידי LAN
7	יישום	Application	MS Network / Dos 3.1 מערכת הפעלה
6	תצוגה	Presentation	NetBios – תוכנת תצורה לתקשורת
5	שיחה	Session	
4	משלוח	Transport	Xerox XNS ,IEEE 802
3	רשת	Network	IEEE 802.3 ,IEEE 802.4 IEEE 802.5
2	קו תקשורת	Data Link	
1	פיסית	Physical	

9.3.1 תקני רשתות LAN

802.3 – תקן לרשת הפועלת בשיטת CSMA/CD כמו Ethernet, למשל.
802.4 – תקן לרשת TOKEN BUS.
802.5 – תקן לרשת TOKEN RING.

בהמשך נסביר כל אחד מהתקנים האלה.

9.4 המודם – MODEM

9.4.1 מהו מודם?

המודם – MODEM (Modulator / Demodulator) הינו מכשיר אשר מתווך בין המחשב הפועל השיטה דיגיטלית לבין מערכת התקשורת, אשר פועלת בשיטה אנלוגית. כלומר, כל רשת תקשורת שאינה פועלת בשיטת פס-בסיס מחייבת שימוש במודם, אשר מתאים לאיפיוני הרשת. המחשב מעביר למודם, או מקבל ממנו נתונים בייצוג ספרתי דיגטלי, בשיטת פס-בסיס. המודם "מדבר" עם רשת התקשורת שאליה הוא מחובר באמצעות שני תדרים (שני צלילים): צליל אחד לשידור נתונים וצליל אחד לקליטת נתונים.

המודם נראה בדרך כלל כקופסה מלבנית, אשר שלושה פתילים יוצאים ממנה. פתיל אחד מיועד לחיבור מקור מתח חשמלי והשנים האחרים מיועדים לנתונים: האחד מוליך למחבר הטורי במחשב והשני – למערכת תקשורת כלשהי. אפשר להשיג מודמים, אשר מורכבים על כרטיס המותקן במחשב האישי. במקרה זה יוצא מהכרטיס פתיל אחד, המתחבר למערכת התקשורת. מערכת התקשורת יכולה להיות לדוגמה קו טלפון של בזק, קו למערכת פרטית, כבל של טלוויזיה בכבלים, קו לרשת תקשורת מחשבים ישראנת של בזק, קו לרשת סיפרנת של בזק וכו'.

המודם יכול להתחבר לקו התקשורת בשתי דרכים:

1. **חיבור 2 Wire:** בחיבור זה קו התקשורת מורכב משני תילים, אשר מהווים קו תקשורת יחיד. תדר השידור ותדר הקליטה מופיעים ביחד על הקו. חיבור 2 Wire אופייני למודם המתחבר לקו הטלפון.

2. **חיבור 4 Wire:** בחיבור זה קו התקשורת מורכב מארבעה תילים. שנים מיועדים לשידור ושנים – לקליטה. בחיבור 4 Wire תדר קליטה מופיע על שני תילים ותדר השידור מופיע על שני התילים האחרים. כלומר, יש הפרדה מוחלטת בין שידור לבין קליטה. חיבור זה אופייני למערכות הקשורות באופן ישיר זו לזו - קשר נקודה לנקודה (נל"ן).

9.4.2 תכונות מודמים

המודמים נבדלים זה מזה בתכונות רבות וכתוצאה מכך, גם במחירים:

* מהירות העברת הנתונים נמדדת בסיביות נתונים לשנייה (BPS) ותלויה גם באיכות קו התקשורת.
* יכולת התגברות על רעשים בקווי התקשורת, גילוי ותיקון שגיאות הנובעות מכך בנתונים המשודרים.
* אפשרות דחיסת נתונים לצורך הקטנת כמות הסיביות המועברות דרך קו התקשורת, כדי להקטין את עלות ההעברה ומשך זמן השידור.

צורת החיבור של המודם למערכת המחשב נקבעת על פי תצורת מערכת המחשב, המרחק בין המחשב האישי למחשב הראשי ואיכות קו התקשורת.

9.4.3 סוגי מודמים

מודם נל"ן

מודם לחיבור נל"ן – PTP (נקודה לנקודה – Point to Point) הוא המודם הבסיסי. כאשר רוצים בתקשורת נל"ן בין מערכות, אשר מרוחקות זו מזו מאות מטרים או מאות ואלפי קילומטרים, אין אפשרות לחבר ביניהן ישירות על ידי חיבור RS232, אשר מוגבל על פי התקן ל-15 מטר.

המודם מוצב סמוך למחשב ומחובר אליו בקו RS232 והעברת הנתונים ביניהם היא בייצוג ספרתי. המודם מחובר דרך קו נל"ן למודם הנמצא ליד המחשב בצד השני ומעביר לו את הנתונים בייצוג אנלוגי. אורך קו נל"ן האנלוגי תלוי בטיב קו התקשורת, טיב המגברים לאורכו וסוג המודם והוא יכול להגיע לאלפי קילומטרים. כדי לספק שירות לטווח גדול מותקנים בו מגברים ומתאמים שונים.

מודם לקו טלפון

מודם לקו טלפון הוא מודם נל"ן מסוג 2 Wire. הוא תואם גם לתקני רשת הטלפון ומסוגל לבצע התקשרות בחיוג דרך הרשת. כאשר המחשב רוצה ליצור קשר, הוא שולח למודם צירוף מיוחד של תווים, המודם עובר למצב חיוג והרצף של תווי ASCII שיישלח אליו הוא המספר שיחייג.

יש לוודא שהמודם יודע לבצע חיוג מתקפים (Pulses), או צלילים בהתאם לסוג החיוג הנדרש בקו הטלפון. כמו כן, יש לוודא שהמודם יודע לזהות צלילי חיוג לפי התקן של חברת בזק.

כיום רבים יותר השירותים הציבוריים הניתנים באמצעות מודם חיוג, כמו לדוגמה שירותי בנקים, שירותים של מאגרי מידע (עורק זהב, שירות מודיעין 133 של בזק לאיתור מספר טלפון ועוד).

מודם למערכת טלוויזיה בכבלים

עם כניסת הטלוויזיה בכבלים יוכלו החברות המפעילות את הכבלים לאפשר חיבור מודמים לרשת הכבלים בערוצים מיוחדים המוקצים לכך במקביל לשידורי הטלוויזיה. קיימים היום בשוק מודמים המסוגלים לספק שירותי תקשורת נל"ן, שירותי מיתוג ואפילו העברת שיחות טלפון במקביל להעברת הנתונים (כמו למשל, חברת Phase Com).

9.5 המחשב האישי כתחנת עבודה למחשב מרכזי

מחשב אישי יכול לשמש כתחנת עבודה, או מסוף למחשב מרכזי. בסביבת עבודה זו יש צורך לשלב בין מסוף המשמש כיחידת קצה למחשב המרכזי לבין הצורך בתחנת עבודה עצמאית בעלת כושר עיבוד עצמאי. המחשב האישי ממלא תפקידים אלה בדרך טובה.

בדרך עבודה זו נבדיל בין שתי משפחות של מחשבים מרכזיים:

1. **מחשבים שהחיבור אליהם נעשה בתקשורת טורית אסינכרונית,** כמו בסביבת עבודה של מחשבי דיגיטל, דטה ג'נרל ועוד. בנוהל תקשורת זה יש צורך בתכנית תקשורת, אשר מדמה את המחשב האישי למסוף של המחשב המרכזי. תכניות תקשורת שונות מאפשרות הדמייה של מסופים אסינכרוניים, כמו למשל SmartTerm, Mirror ועוד. יש לבחור את תכנית ההדמיה על פי סוגי המסופים שיש ביכולתה לדמות וסוגי המסופים הנדרשים לעבודה מול המחשב המרכזי. תכנית הדימוי גורמת לכך שהמחשב האישי פועל בתיאום מלא עם המחשב המרכזי כאילו היה מסוף.

   התקשורת מתבצעת על ידי חיבור של קו נתונים בין כרטיס התקשורת האסיכרונית הנמצא במחשב לבין המחשב המרכזי. החיבור בין המחשב האישי לבין המחשב המרכזי נעשה באמצעות ערוץ תקשורת פשוט וזול המכיל 3 חוטים בלבד (ראה פרק על תקשורת טורית). החיסרון העיקרי של תקשורת טורית אסיכרונית הוא בחוסר האמינות שלה. פרוטוקול התקשורת אינו תומך בגילוי ותיקון שגיאות ולכן קיימת בעיה גדולה בהעברת קבצים בצורה אמינה בין המחשבים.

2. **מחשבים שהחיבור אליהם נעשה בתקשורת טורית סינכרונית.** שיטה זו מאפיינת בעיקר סביבת עבודה של מחשבי חברת יבמ, או מחשבים תואמים. בשיטה זו דרושה תוכנת הדמיה, כמו Irma, Pcox Linkup, ובכרטיס תקשורת סינכרונית. הכרטיס לתקשורת סינכרונית יקר יותר מכרטיס

תקשורת טורית אסינכרונית. במרבית המחשבים האישיים לא נמצא כרטיס לתקשורת טורית סינכרונית.

החיבור בין המחשב האישי לבין המחשב המרכזי נעשה בעזרת כבל קואכסיאלי המקשר בין הכרטיס הסינכרוני במחשב לבין בקר התקשורת של המחשב המרכזי. כיום ישנם התקנים המאפשרים להמיר את הכבל הקואכסיאלי בזוג חוטי טלפון וכך לחסוך בצורה משמעותית בעלות הכבל. בפרוטוקול תקשורת סינכרונית קיימת בקרת שגיאות טובה ואפשרות לתיקון שגיאות ולכן העברת הנתונים אמינה יותר.

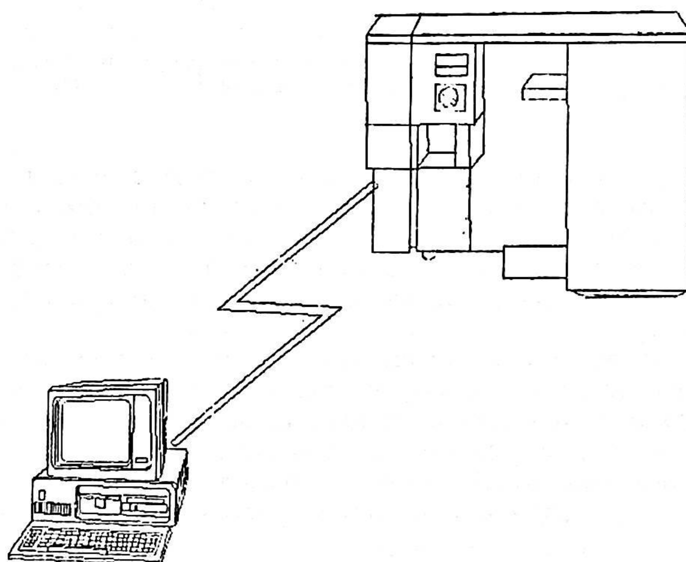
9.5.1 המחשב האישי כמסוף למחשב מרכזי

היתרון המשמעותי בחיבור של המחשב האישי כמסוף למחשב מרכזי על פני חיבור של מסוף רגיל הוא ביכולת העיבוד העצמאית שלו. בעזרת תוכנת התקשורת ניתן להשתמש במחשב האישי כמסוף וגם להעביר קבצים בין המחשבים השונים כדי לבצע את עיבוד הנתונים במחשב האישי ולא במחשב המרכזי. עיבוד במחשב האישי מוריד מהעומס על המחשב המרכזי ומנצל את היתרונות הגרפיים שלו, בצד המחיר הנמוך יחסית של תכניות העיבוד.

תוכנות התקשורת מאפשרות:

- שליטה על הפרמטרים הקשורים בתקשורת, כמו מהירות, מספר סיביות נתונים וכו',
- דימוי של המחשב האישי לסוגי מסופים שונים,
- העברת קבצים בין המחשבים,
- אחסון נתונים במערכת הדיסקים שבמחשב האישי,
- המרת קודים ASCII ל-EBCDIC ולהיפך,

ועוד.

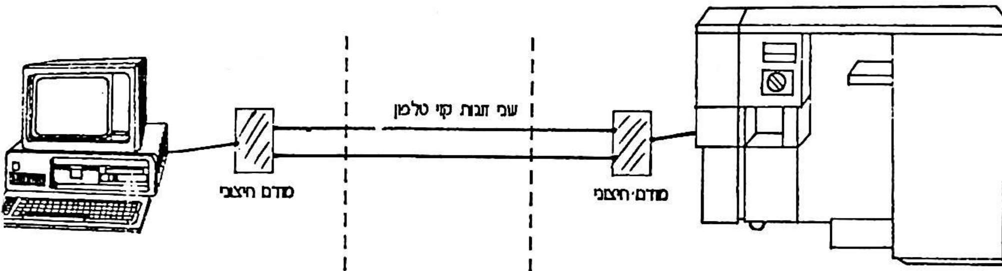


חיבור PC כמסוף למחשב מרכזי

9.5.2 חיבור מחשב אישי למחשב מרוחק

כאשר המחשב האישי מרוחק מהמחשב המרכזי יכולים להיות שני מצבים: אי אפשר לחבר ישירות בינו לבין המחשב המרכזי באמצעות תקשורת טורית אסינכרונית בגלל המרחק, או שאי אפשר לחבר ישירות בינו לבין הבקר של המחשב המרכזי בתקשורת סינכרונית (במערכות יבמ). תקן RS232 מתיר מרחק עד 15 מטרים, אם כי באופן מעשי ניתן להגיע במקרים מסוימים עד למאות מטרים, אך אין זה מומלץ. במקרים אלה יש צורך לחבר את המחשב האישי אל המחשב המרכזי או אל בקר התקשורת של המחשב המרכזי באמצעות מודם, אשר מאפשר ליצור קשר למרחקים גדולים.

יש להזכיר כאן, שישנם שערי גישור בין רשתות מקומיות לבין מחשבים מרכזיים. הם מאפשרים לכל תחנה ברשת המקומית לפעול כמסוף מול המחשב המרכזי.



חיבור PC כמסוף למחשב מרוחק (שימוש במודם)

9.6 רשתות תקשורת מקומיות (LAN)

רשתות תקשורת מקומיות (LAN - Local Area Network) נועדו לחבר בין מחשבים אישים. הן מאפשרות שיתוף משאבים והעברת נתונים בין מחשבים שונים הנמצאים באותו אתר. הן יכולות לתת פתרון בעיקר ברמת הארגון, כלומר פתרון לקשר בין מחשבים הנמצאים במרחק קצר זה מזה באותו בנין, או באותו אתר ועם צרכים ומטרות משותפים.

רשתות התקשורת המקומיות השונות נבדלות זו מזו בשירותים שהן מעניקות למשתמשים ברשת ובצורת החיבור בין המחשבים השונים. המרחק תלוי בסוג הרשת ולא עובר טווח של קילומטרים ספורים. רשתות התקשורת הראשונות נתנו פתרון של חלוקת משאבים ושיתוף משאבים מרכזיים בין תחנות העבודה השונות, כמו מדפסות, דיסקים מרכזיים וכו'. גם בעת שיתוף ברשת נשמר האופי של המחשב האישי כתחנת עבודה למשתמש יחיד.

רשתות מקומיות מאפשרות חיסכון במשאבים והן כדאיות עקב מחירם הגבוה של ההתקנים ההיקפיים: דיסקים, טייפים לגיבוי, מדפסות מהירות וכו'. עיקר

הבעיה של רשתות אלו היא בשיתוף הנתונים בין המחשבים. השיתוף לצורכי קריאה בלבד הוא זמין ויעיל. כלומר, אם הצורך היחידי הוא לשאוב נתונים ממאגר מידע מרכזי, טעינה של תכנית מדיסק מרכזי ופעולות נוספות הכרוכות בקריאה בלבד, הרי שרשת התקשורת מספקת את כל הכלים הדרושים.

עם זאת, רשתות התקשורת לא סיפקו בתחילה כלים יעילים לצורך עדכון של אותו מאגר נתונים על ידי משתמשים אחדים. כלומר, אם שני מחשבים צריכים היו לעדכן את אותו קובץ נתונים הרשת לא סיפקה כלים מתאימים לצורכי נעילה הדדית של קבצים, רשומות או שדות, אשר בלעדיהם לא ניתן לאפשר לשני מחשבים, או יותר, לגשת ולעדכן את אותם נתונים. כלומר, רשתות תקשורת אלו הציבו מגבלות רבות ליישומים שדרשו שיתוף נתונים ומשאבים באופן מלא לפעולות קריאה או כתיבה.

רשתות התקשורת הקיימות כיום פותרות את הבעיות לשיתוף משאבים ונתונים, תוך שימוש בכלים מתאימים לצורכי נעילה מפני עדכון בו-זמני ממספר מחשבים. הפתרונות התאפשרו עקב ההכרזה על **תקן NetBios**, המאפשר גישה לנתונים ברשת בצורה תקנית ואמינה באמצעות מערכת ההפעלה. כמו כן בוצעו שינויים במערכת ההפעלה DOS החל מגירסה 3.1 בצורה שאפשרה לה לתמוך בעבודה ברשת תקשורת.

עד ליישום של הרשתות המקומיות החדשות בארגון היו המחשבים האישיים נפרדים לחלוטין משאר מערכות המחשוב. עתה הם החלו לנגוס בנתחי שוק של מחשבי מיני ואפילו של מחשבים גדולים. יש לזכור שטיפול במערכות של בסיסי נתונים גדולים כמו חשבונות בבנק או ביצוע חישובים מורכבים כמו תחזית מזג אוויר, מאגרי מידע גדולים המשמשים מספר גדול מאוד של משתמשים, עדיין רחוקים מיכולתו של המחשב האישי, גם כאשר הוא פועל ברשת תקשורת מקומית וגם כאשר עוצמתו רבה.

לרשתות התקשורת המקומיות יתרונות גדולים על המחשב המרכזי. צורת עבודה בהן הינה בשיטת **עיבוד מבוזר** (Distributed Processing), לעומת גישת העיבוד הריכוזי של המחשב המרכזי. חיבור ברשת מאפשר לכל מחשב "לשוחח" עם כל מחשב אחר ברשת ועם זאת, אינו מבטל את יכולת העבודה העצמית של כל תחנה ברשת. עיבוד הנתונים נעשה בכל מחשב בפני עצמו ולא במערכת המרכזית ועל כן, "נפילה" של תחנה כלשהי ברשת אינה גורעת מיכולת העבודה של יחידה אחרת. כמובן, פרט לשירותים ולנתונים הנמצאים בתחנה שאיננה פעילה באותו זמן ואינם זמינים לתחנות אחרות.

שיתוף נתונים בתצורה של רשת תקשורת מספק מספר יתרונות:

1. אי כפילות נתונים, מכיון שיש צורך בעותק יחיד של תוכנה או נתונים בדיסק מרכזי. כל תחנה יכולה להשתמש בעותק המרכזי.
2. זמינות ועדכניות של הנתונים בזמן אמיתי לכל התחנות.
3. גיבוי מרכזי ומניעת אובדן נתונים.
4. העברת נתונים בצורה מהירה ואמינה, לעומת הקצב הנמוך יותר בעת שימוש במחשב אישי כמסוף קצה למחשב מרכזי.
5. אפשרות קישור לרשתות תקשורת רחוקות.
6. בין המחשבים ברשת ניתן לשתף ציוד יקר ובכך לחסוך בעלות הגבוהה של ציוד זה (שיתוף משאבים בין המחשבים).

9.6.1 תצורת רשתות תקשורת וכרטיס מתאם לרשת התקשורת

רשתות התקשורת מיישמות את שתי הרמות הראשונות של מודל OSI:

1. **הרמה הפיסית** (Physical Layer) מפרטת את האותות החשמליים הזורמים בערוץ התקשורת.

2. **רמת ערוץ הנתונים** (Data Link) מפרטת את האופן שבו זורמים הנתונים בערוץ (או קו התקשורת). היא מספקת בקרה על זרימת הנתונים ממקור ליעד ללא תקלות ושגיאות ובכך מבטיחה קו תקשורת אמין.

הרשתות השונות נבדלות ביניהן בכמה אפיונים:

9.6.1.1 תווך התקשורת

תווך התקשורת (Media) הינם הכבלים המרכיבים את רשת התקשורת ואשר בהם עוברים האותות החשמליים. הכבלים הנפוצים הם: כבל קואכסיאלי 70 אום או 50 אום, כבל שזור או שטוח, חיבור 4 Wire או חיבור 2 Wire. סוג הכבל קובע את מהירות ההעברה המירבית של הנתונים בערוץ התקשורת. למשל, כבל עם זוג שזור מסוכך מאפשר מהירות העברה של כ-10MBit/Sec, ולעומתו כבל קואכסיאלי מאפשר העברת נתונים עד כ-300MBit/Sec למרחק רב יותר, אך מחירו גבוה יותר.

9.6.1.2 תצורת רשת התקשורת (Network Topology)

אנו מכירים בכמה תצורות רשת תקשורת:
ערוץ משותף – Bus, טבעת – Ring, כוכב – Star.

תצורת ערוץ משותף — Bus

בתצורת חיבור של ערוץ משותף (ראה תרשים) כל התחנות קשורות לערוץ תקשורת דו כיווני הפרוש באתר. שני קצוות ערוץ התקשורת אינם מתחברים ביניהם ולכן יש להתקין בהם רכיבי סיום (Terminators) על מנת לשמור על התכונות הפיסיות של הערוץ. רכיב הסיום הוא רכיב חשמלי שנועד לאפשר תיאום של עכבות סיום (Impedance Matching) לקו, בקצוות שאליהם לא מחוברות מערכות מחשב. כל התחנות מחוברות לערוץ התקשורת וכולן מאזינות לנתונים העוברים בו.

יתרונות:
1. חיבור פשוט של כל תחנה לערוץ התקשורת. כל שנדרש הוא ניתוק הקו באמצע וחיבור מתאם T, שמאפשר חיבור של התחנה החדשה לערוץ התקשורת.
2. אורך הכבל הפיסי הנדרש לצורך חיבור התחנות באתר קטנה לעומת חיבורים בתצורות אחרות. משיגים זאת מכיון שניתן לחבר כבל יחיד בין כל התחנות ולחבר כל תחנה לערוץ באמצעות מתאם וכבל קצר.
3. נפילה של תחנה כלשהי ברשת אינה מפילה את כל הרשת.

חסרונות:

1. קשה לגלות נתק בערוץ התקשורת, או כל תקלה אחרת.
2. פרוטוקול התקשורת ליישום התקשורת הדו-כיוונית מסובך.

תצורת טבעת — Ring

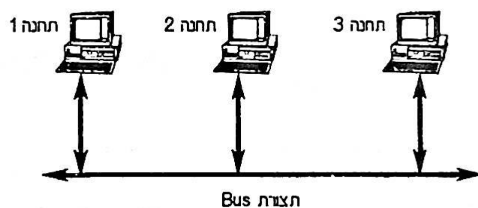
בתצורת חיבור טבעת (ראה תרשים) מחוברות כל התחנות במעגל. ערוץ התקשורת הוא חד-כיווני והנתונים זורמים בו מתחנה אחת לתחנה הבאה. כל תחנה מאזינה לנתונים שתחנה קודמת העבירה אליה.

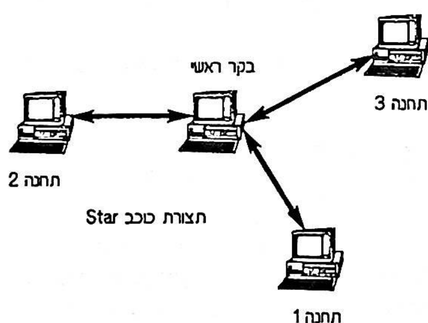
יתרונות:

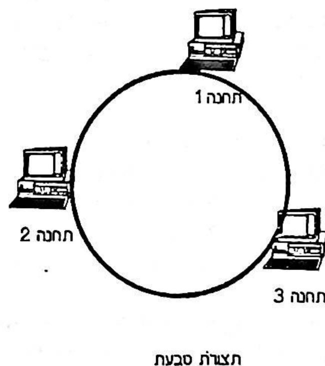
1. קל לגלות בשיטת חיבור זאת נתק בערוץ התקשורת.
2. פרוטוקול התקשורת ליישום תקשורת דו-כיוונית פשוט.
3. נפילה של תחנה כלשהי ברשת אינה מפילה את כל הרשת.

חסרונות:

1. מסלול זרימת הנתונים ברשת ארוך יותר מכיון שכל תחנה מייצרת מחדש את האות הזורם בערוץ התקשורת ומעבירה אותו לתחנה הבאה.
2. חיבור של תחנה חדשה לערוץ כרוך בסלילת כבל תקשורת נפרד ולכן אורך כבלי ערוץ התקשורת גדול יותר מחיבור בתצורת Bus. לדוגמה, בפריסת רשת תקשורת Token Ring של יבמ כל תחנה ברשת מתחברת בעזרת כבל תקשורת למרכזיית פתילים, אשר יוצרת את הטבעת.







תצורות (Topology) של רשתות תקשורת מקומיות

תצורת כוכב — Star

בצורת חיבור כוכב (ראה תרשים) כל התחנות ברשת מקושרות לתחנה מרכזית אחת המנתבת את הנתונים הזורם ברשת.

יתרונות:

1. קל ונוח להוסיף תחנות חדשות לרשת.
2. איתור מהיר של תקלות בתחנות.
3. מעגלי חומרה פשוטים ביותר של כרטיסי הרשת.

חסרונות:

1. נפילה של תחנת שירות מרכזית תגרום לנפילה של כל רשת התקשורת. מצב קריטי זה יש למנוע עד כמה שניתן בעבודה מבוזרת ברשת תקשורת.
2. עומס גדול על תחנה מרכזית כאשר מחברים מספר רב של תחנות קצה.
3. כל תחנה דורשת כבל נפרד למאגד המרכזי.

9.6.1.3 פרוטוקול התקשורת ברשת

כיצד מנהלת רשת התקשורת את מעבר הנתונים ברשתות Token Ring, CSMA/CD ואחרות?. פרוטוקול התקשורת מפרט כיצד עוברים הנתונים ברשת ברמת שכבת הקו (Data Link Layer) של מודל OSI. הפרוטוקול מאפשר גילוי ותיקון שגיאות בהעברת הנתונים על מנת לאפשר זרימה אמינה שלהם בין התחנות השונות המחוברות ברשת.

שני פרוטוקולים עיקריים נמצאים בשימוש ברשתות תקשורת:

פרוטוקול CSMA/CD
(Carrier Sense Multiple Access / Collision Detect)

בפרוטוקול CSMA/CD מחוברות כל התחנות על ערוץ תקשורת יחיד ולכל אחת מהן ישנו קוד גישה (כתובת). כל מסר העובר בערוץ מכיל את תחנת היעד, תחנת המקור, נתוני בקרה לגילוי ותיקון שגיאות ונתונים המיועדים לתחנת היעד. כל התחנות מאזינות למסרים הזורמים בערוץ על מנת שתוכלנה לזהות את אלה המיועדים להן ולקלוט אותם.

כל תחנה ברשת שמעוניינת לשדר נתונים חייבת לתפוס שליטה בלעדית על ערוץ התקשורת, כי רק תחנה אחת יכולה לשדר בערוץ בו-זמנית. התחנה היוזמת התקשרות מאזינה לערוץ ומסוגלת לזהות אם קיימת בו זרימת נתונים (Carrier Sense). כל התחנות יכולות ליזום התקשרות (Multiple Access) באותו זמן. כל אחת מהן תנסה לתפוס שליטה על ערוץ התקשורת ולשדר דרכו נתונים, רק אם הרגישה שאין בו שידור של תחנה אחרת, כי רק תחנה אחת יכולה לשדר נתונים בצורה אמינה דרך הערוץ בעת ובעונה אחת. שידור במקביל של מספר תחנות יגרום לשיבוש הנתונים. בפרוטוקול זה כל התחנות הן במעמד שווה ואין תחנה אחת המרכזת את הבקרה ולכן נפילה של תחנה ברשת אינה גורמת לנפילת הרשת.

כאשר תחנה אחת בלבד מנסה לשדר, הנתונים שהיא משדרת יעברו בצורה תקינה בערוץ ויגיעו ליעדם. יכול להיות מצב שבו כל התחנות יוזמות התקשרות

ומגלות שאין זרימת נתונים בערוץ ומנסות לשדר בו-זמנית. במצב זה קורית התנגשות נתונים, אשר גורם לשיבוש שלהם. הכרטיס המתאם לרשת בכל תחנה יכול לזהות התנגשות כזו (Collision Detect), לבטל את ניסיון השידור ולנסות שנית.

על מנת למנוע אפשרות שכל התחנות ינסו לשדר שוב באותו הזמן לאחר הכישלון, כל תחנה בוחרת מספר אקראי ומנסה לשדר אחרי פרק הזמן האקראי שבחרה, כדי להוריד למינימום את סיכויי ההתנגשות. כאשר הרשת אינה עמוסה, סיכויי ההתנגשות קטנים ופרק הזמן המבוזבז על נסיונות שידור סרק הוא מזערי. ככל שהרשת עמוסה יותר ומספר התחנות עולה, גם סיכויי ההתנגשות גדלים והזמן המבוזבז על ניסיונות סרק עולה.

מספר תוכנות לניהול רשת תקשורת משתמשות בפרוטוקול חומרה זה, כמו Novell למשל. הפרוטוקול CSMA/CD מיושם בדרך כלל בתצורה של Bus.

קצב העברת הנתונים ברשת Ethernet המשתמשת בפרוטוקול זה הינו כ-10 מיליון סיביות לשנייה. החיבורים ברשת מיושמים על ידי כבל קואכסיאלי בן 75 אום, אשר זהה לכבלים של אנטנות טלוויזיה עם מחברי BNC. כדי להגדיל את טווח השידור מכניסים מגברים עד למרחק מקסימלי של כ-3 ק"מ.

כל מסר (Message, או הודעה) העובר ברשת כולל מספר מרכיבים על פי הסדר הבא:

כותרת המסר (Header)	- 8 בתים ראשונים של המסר המאפשרים סינכרון בין המערכת המשדרת לקולטת.
כתובת היעד	- 6 בתים המזהים את הנמען.
כתובת מקור המסר (השולח)	- 6 בתים המזהים את השולח.
סוג המסר המועבר	- 2 בתים המודיעים על סוג המסר: אם זהו מסר נתונים, אישור על מסר, הודעה על שגיאות בקליטה וכו'.
גוף המסר	- עד 1500 בתים המכילים את תוכן המסר.
בקרת שגיאות (CRC)	- 4 בתים המכילים ערך המשתנה על פי תוכן המסר ומאפשרים זיהוי שיבושים בהעברה של הנתונים בין התחנות.

פרוטוקול CSMA/CD לרשתות הוא הנפוץ ביותר גם ברשתות תקשורת למחשבים גדולים בשל הפשטות בחיבור הרשת ומחיר הקמתה הנוח. רשת Ethernet ורשתות דמויות לה, כמו Cheapernet הן הרשתות הנפוצות ביותר. הן משתמשות בפס-בסיס (BaseBand) וכוללות אפנון לוגי של השעון והנתונים.

פרוטוקול Token Ring

פרוטוקול **רשת האסימון**, או **רשת הסמן** (Token Ring) מאפשר חיבור של רשת טבעתית (Ring). כל התחנות ברשת קשורות זו לזו באופן שכל תחנה מקבלת נתונים מהתחנה שלפניה ומשחזרת אותם עבור התחנה שנמצאת אחריה ברשת. כדי לפקח על השידור ברשת מגדירים סדרה של סיביות רצופות הנקראת אסימון, או סמן (Token), אשר עובר בין תחנה לתחנה, אך נמצא בו-זמנית בתחנה אחת בלבד.

ברגע שתחנה כלשהי מקבלת ומזהה אסימון (קוד ASCII מסוים) ומעוניינת לשדר נתונים, עליה להוריד אותו מהרשת על מנת שהתחנה הבאה אחריה לא תקבל אותו ותשדר גם היא. לאחר שתחנה הורידה את האסימון היא משדרת את הנתונים שברצונה לשדר ולאחריהם היא מחזירה את האסימון אל ערוץ התקשורת.

פרוטוקול העברת אסימון יוצר סדר בגישה של תחנות לרשת, כי הוא מאפשר לתחנה אחת בלבד לשדר בזמן כלשהו ומונע בזבוז זמן בניסיונות שידור סרק. לפרוטוקול העברת אסימון יתרון גדול ברשת עמוסה, אך כאשר הרשת אינה עמוסה מתבזבז זמן בהמתנה לאסימון שעובר ברשת בתנועתו בין תחנות שאינן רוצות לשדר.

רשת העברת אסימון מנוצלת בתוכנות ניהול רשת כמו Corvus ו-Novell.

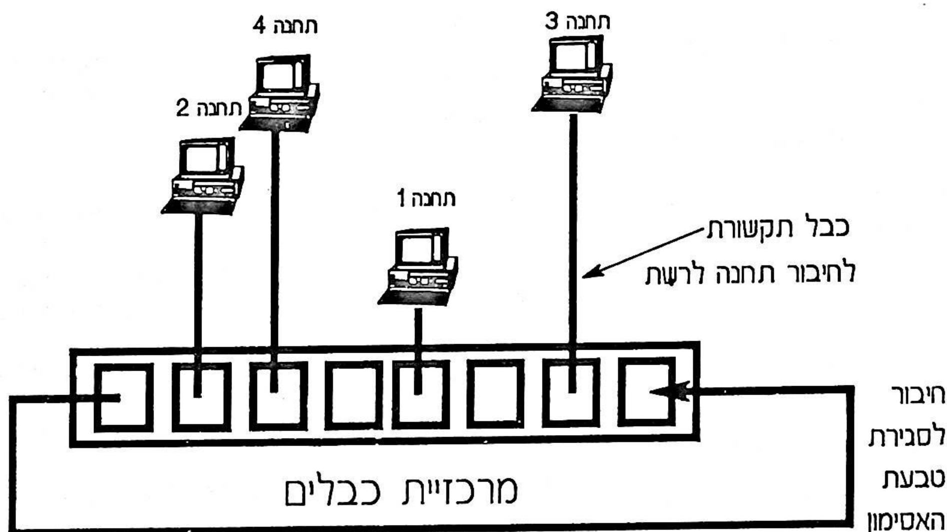
החיבור הפיסי דומה לחיבור כוכב כשבמרכזו מרכזיית הרשת, שבה מופעל עיקרון הטבעת. כפי שציינו למעלה, הוכרז על ידי יבמ שער גישור (Gateway) בין 802.5 לבין 802.3 כלומר, בין רשת Token Ring של יבמ לבין רשת Ethernet, בהתאמה. קצב העברת הנתונים הוא עד 16Mbit/sec (מיליון סיביות לשנייה) ברשת Token Ring של יבמ. רשת יבמ Token Ring היא השנייה בתפוצה.

שני יתרונות בולטים לרשת זו:

1. היא מיועדת לעבודה בתנאי עומס תקשורת.
2. היא נתמכת על ידי יבמ.

חסרונותיה:

1. סיבוכיות גבוהה יותר של תצורת החיבור של המערכות לרשת.
2. מחיר יקר יותר של רשת התקשורת.



רשת Token Ring בשיטת הכבלים של יבמ
(תצורת הפריסה של הרשת)

9.6.1.4 חיבור המחשב לרשת

כדי לחבר את המחשב האישי לרשת תקשורת בתצורת תקשורת כלשהי דרושים מספר רכיבים חיוניים:

1. כרטיס תקשורת לחיבור המחשב לרשת. כרטיס התקשורת חייב להיות מותאם לסוג רשת התקשורת. הוא מיישם את פרוטוקול התקשורת ומאפשר לתוכנת הרשת להשתמש בו כדי לספק את השירותים הנדרשים.

2. כבל תקשורת שמחבר בין כרטיס התקשורת במחשב לבין הרשת.

3. תכנית המאפשרת למחשב האישי שירותים בסיסיים להעברת נתונים דרך כרטיס התקשורת שבמחשב אל רשת התקשורת. תכנית זו יכולה להמצא על גבי כרטיס התקשורת ברכיב Boot ROM המבצע אתחול וטעינה לזיכרון בשלב Boot של המחשב האישי, או באמצעות תוכנת Device Driver הנטענת בעת טעינת קובץ CONFIG.SYS.

4. תכנית רשת תקשורת המאפשרת לבצע את הפעולות השונות ברשת, כמו הדפסה, טעינת תכניות והעברת מסרים. קיימים סוגים שונים של תכניות לרשתות תקשורת כמו: Novell, Corvus, IBM PC LAN ועוד.

9.6.2 מה מקבל המשתמש מהרשת?

תכנית רשת התקשורת מאפשרת לתחנות השונות להשתמש בחומרה (כרטיסי תקשורת וכבלים) לצורך העברת נתונים בין התחנות ומתן שירותים שונים.

9.6.2.1 שירותי רשת התקשורת

* **שיתוף קבצים ודיסקים:** מספר תחנות פונות לדיסק מרכזי. מושג חיסכון בעלויות דיסקים ובכפילות תכניות.

* **העברת קבצים:** שירותים נוחים להעברת קבצים בין תחנות.

* **חסימה ונעילה של קבצים/רשומות/שדות:** החסימה חיונית להקמת מאגרי מידע ובסיסי נתונים. כאשר מספר תחנות ניגשות לאותם נתונים ורוצות לשנות אותם, צריך לחסום את הגישה מתחנות אחרות כל עוד התחנה הכותבת לא סיימה לעשות זאת. הרשת צריכה לספק אפשרות נעילה ושחרור של גישה לנתונים ברמות שונות: קבצים, רשומות, שדות. חסימה באופן זה מאפשרת חסימה על פי שכבות נתונים, זמנים, סיסמאות ועוד.

* **סיסמאות וקודים לחסימה** של משתמשים לא מורשים.

* **שיתוף מדפסות:** אפשרות שימוש במדפסת אחת למספר מחשבים.

* **אמולציה של מסופים:** הרשת מאפשרת שער גישור (Gateway) למחשב מרכזי ומספקת אפשרות חיקוי של תחנה ברשת למסוף של מחשב מרכזי.

* **דואר אלקטרוני, מיתוג מסרים:** אפשרות העברת מסרים בין תחנות. דואר אלקטרוני פוטר את המשתמש מהצורך בהעברת תכתובת ידנית.

* **ניהול רשת התקשורת:** ניהול של מערך החשבונות במחשב, כמו זמנים, עלויות שימוש ועוד. שמירת מידע על שגיאות במערכת, הוספת דיסקים לרשת והורדתם, מתן מידע על ביצועי הרשת ומצבה (Status, Load, Users, Open Files) ועוד.

* **רישום זהה על שני דיסקים במקביל** להעלאת אמינות הנתונים ואבטחתם.

* **סגירת הרשת בצורה מבוקרת** ולא הרסנית במידה ונפסקה אספקת החשמל ומכשיר האל-פסק (UPS) החל לפעול.

* **הרצה של יישומים על השרת** (Server) מקנה מהירות הרצה גבוהה יותר. אפשרי במספר רשתות על שרתים 386 ומעלה.

במרבית הרשתות המשתמש מקבל מהרשת את שלושת השירותים הראשונים הנזכרים בסעיף הקודם. מימוש הרשת נעשה באמצעות הגדרת דיסק לוגי (ראה סעיף 3.3.6) המסומן לרוב באות F (אפשר לחבר דיסקים נוספים). עבור משתמש או תכנית היישום הדיסק הלוגי הוא דיסק לכל דבר, יש בו ספריות וקבצים, אפשר לקרוא ולהעתיק קבצים עליו וממנו. ההבדל הוא בכך שהדיסק הלוגי אינו נמצא על המחשב הנוכחי, אלא במחשב אחר שקשור למחשב של המשתמש באמצעות רשת התקשורת. כלומר, דיסק במחשב שכלשהו ברשת, אשר משרת גם אחרים מוגדר אצלם כדיסק F. במרבית רשתות התקשורת קיים מחשב אחד שעוצמתו גבוהה מזו של חבריו ברשת ובו מותקן דיסק שקיבולתו גבוהה. זהו **שרת הרשת** (Network Server), שתפקידו לעיתים גם ניהול הרשת.

9.6.2.2 מערכות שרת/לקוח (Client/Server)

במספר רשתות (לדוגמה Novell או LAN Manager), יש צורך במחשב שרת בעל דרישות מינימום לחומרה: מעבד 80286, או אף 80386. ברשתות אלו ניתן לחבר יותר משרת יחיד. שאר התחנות ברשת יכולות להיות מחשבים מכל סוג שהוא ללא מגבלה.

ברשתות שבהן פועל **שרת** (Server), כל **לקוח** (Client), **תחנה** (Work Station) או שרת אחר יכולים לשאוב נתונים ולקבל שירותים משרת כלשהו ברשת, אך אין אפשרות לשאוב נתונים, או לקבל שירות ממחשב המוגדר **כתחנה** (Work Station). כלומר, לא ניתן לשאוב נתונים או להשתמש במשאבים המורכבים על מחשב חבר ברשת (יע"מ, מדפסת, דיסק וכו'). ברשתות אלו נפילה של השרת תורגש מיידית ותפגע בעבודת הרשת.

ברשתות אחרות (לדוגמה Lantastic או Smartlan) אין מגבלת שרת וכל תחנה יכולה לשמש כשרת לתחנות האחרות, לשתף את משאביה ולהשתתף במשאבי התחנות האחרות. כלומר, כל מחשב ואפילו PC/XT יכול לשמש כשרת למטרות מסוימות. השירותים שמספקות רשתות מסוג זה מצומצמים יותר, אך התקנתן פשוטה יותר ונוחה יותר. הן מתאימות למספר קטן של תחנות שמשימותיהן פשוטות.

השיתוף של שרת הדיסק דורש הסברים נוספים:

נניח שיש שני פקידים בבנק, אחד עם מחשב א' והשני עם מחשב ב'. פקיד א' מעדכן את כתובתו של פלוני ופקיד ב' מעדכן באותו זמן את כתובתו של אלמוני. שניהם מעדכנים למעשה את קובץ הכתובות שנמצא בשרת. הבעיה היא בכך שמערכת ההפעלה במחשב השרת חייבת לפתוח את קובץ הכתובות לכתיבה עבור שני משתמשים. פתיחת קובץ עבור יותר ממשתמש אחד יכולה להתבצע מגירסה 3 של DOS. הבעיה שהצגנו קודם מחריפה, כאשר שני הפקידים מעוניינים לעדכן את אותה רשומה. במקרה זה צריך לנעול את הרשומה כאשר הראשון שביניהם מעדכן אותה. בגמר הכתיבה של המשתמש הראשון תשוחרר הרשומה ותהיה זמינה לעדכון על ידי המשתמש השני.

שני מנגנונים מאפשרים ניהול ונעילה:

* פסיקת השירות 21HEX של מערכת ההפעלה DOS. פונקציה 5CHEX של פסיקת השירות מאפשרת נעילת רשומות.

* תכנית SHARE.EXE (מופעלת לרוב ב-Autoexec) מאפשרת שיתוף קבצים על ידי מספר תחנות. תכנית SHARE מוסיפה אפשרויות לאופן פתיחת הקובץ באמצעות מערכת ההפעלה DOS:

<u>בית MODE</u>

משמעות הסיבית	ערך סיביות	מספרי סיביות	
READ	000	0-2	
WRITE	001		
READ/WRITE	010		
Reserved	0	3	
Compatibility	000	4-6	תוספת לשיתוף קבצים באמצעות SHARE
Deny all	001		
Deny Write	010		
Deny Read	011		
Deny None	100		

9.6.3 סיכום

נסקור את פרוטוקול התקשורת במחשבים אישיים:

* אם אתה יצרן של כרטיס תקשורת, אתה:

1. בונה מערכת חומרה שתומכת בתקשורת פיסית מסוג Token Ring Ethernet וכו' (כפי שמפורט בתחילת הפרק), או מסוג אחר.

2. מפיק באמצעות כרטיס רשת התקשורת פסיקת חומרה Irq4 או פסיקה משנית Irq3 (אפשר לבחור בין הפסיקות במתגים שעל הכרטיס).

3. מכין תכנית ROM, הקרויה NetBIOS, אשר מופעלת על ידי פסיקות החומרה. תכנית זו אוספת את הנתונים והפקודות המועברים בערוץ התקשורת של הכרטיס ומפיקה פסיקת תוכנה 44Hex. תכנית ROM מקבלת פקודות ונתונים מפסיקת תוכנה 7AHex ומעבירה אותן לערוץ התקשורת של הכרטיס. כדי לקבל את פרוטוקול המלא לפסיקות התוכנה הזו ראה MS Networks Technical Refernce.

* אם אתה כותב תכנית תקשורת, אתה:

4. כותב תכנית ממשק להתקן תקשורת. תכנית הממשק מקבלת פקודות ממערכת ההפעלה DOS ומגדירה עבורה דיסק לוגי F, או דיסק לוגי אחר. מערכת ההפעלה פונה באמצעותו לרשת התקשורת. ראה הוראות לכתיבת Block Device Driver
בספר MS DOS Programmers Reference.
התכנית מקשרת את הדיסק הלוגי לערוץ התקשורת דרך תכנית NetBIOS באמצעות פסיקות התוכנה 44Hex ו-7AHex.

* אם אתה כותב תכנית ישום שאמורה להתקשר לתכניות ישום נוספות דרך רשת התקשורת:

5. התכנית צריכה לפעול עם נעילת רשומות ושיתוף קבצים בצורה שהסברנו בסעיף הקודם.

שים לב: הצגנו כאן את חמש שכבות של התקשורת על פי מודל OSI, החל מהשכבה הפיסית שמצוינת במספר 1 ועד רמת השיחה המסומנת במספר 5.

9.7 רשתות תקשורת ארציות (WAN)

רשתות תקשורת ארציות שונות באופן מהותי מרשתות תקשורת מקומיות (LAN). רשתות LAN פועלות בקצב של 16MBit/Sec ורשתות ארציות פועלות בקצב 9.6KBit/Sec ועד 2MBit/Sec. כאשר מעוניינים בהעברת נתונים למרחקים גדולים קיימות מספר אפשרויות ליישום הרשת:

מודם טלפון: כפי שצויין, אפשר לחבר שני מחשבים דרך קו טלפון באמצעות יחידות מודם הקשורות בכל קצה. זהו הפתרון הזול ביותר, אבל חסרונותיו רבים. טיב קווי הטלפון בארץ אינו מאפשר פעולה בקצב של 9600Bit/Sec. על כן משתמשים בקצב נמוך יותר כמו 4800 או 2400 וגם אז בחלק מהקווים עלולות להיות שגיאות רבות.

קו PAD ישראנת באמצעות קו נל"ן או דרך טלפון. רשת ישראנת היא המקבילה ל-Telenet האמריקאית. זו היא רשת מיתוג מנות לפי תקן X25. היא מאפשרת למשתמשי מחשבים אישיים להתחבר אליה בחיבור אסינכרוני ולמחשבים גדולים יותר – בחיבור ישיר בפרוטוקול X.25. לכל מחשב המחובר לרשת ישראנת יש כתובת ברשת.

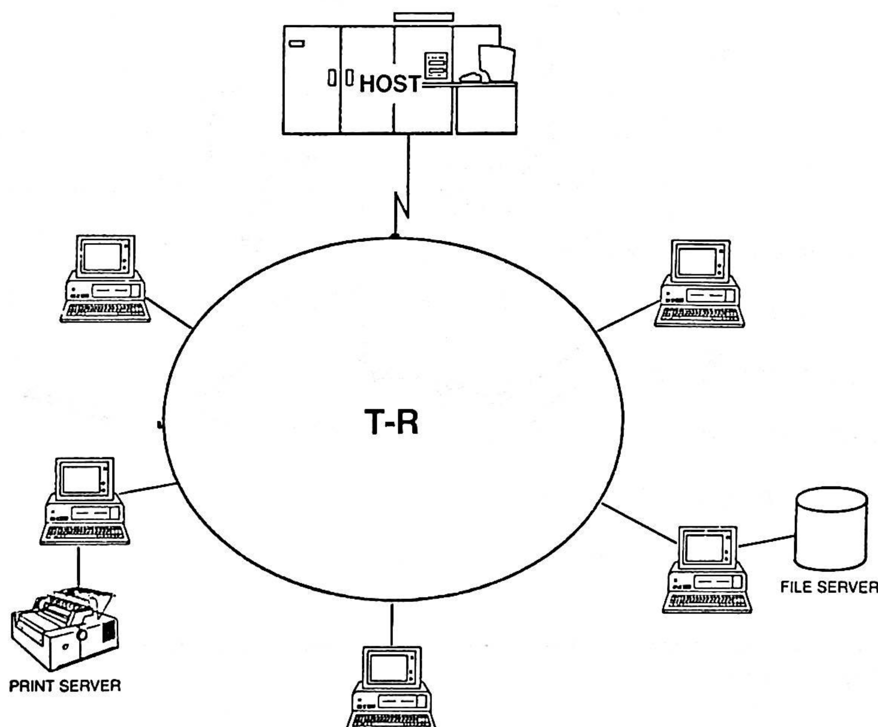
כאשר רוצים ליצור קשר למחשב אחר ברשת, בדומה לפעולת החיוג ברשת הטלפון, מקישים את כתובת המחשב המבוקש והרשת דואגת לחיבור בין

המחשבים. מחשב היעד להתקשרות יכול להמצא בבנין סמוך, בעיר אחרת, או אפילו בארץ אחרת. אמינות העברת הנתונים בתוך הרשת גבוהה יותר מאמינות הקו המחבר בין המחשב לבין הכניסה לרשת בגלל שיטות הבקרה בתוך הרשת.

קו העברת נתונים - קה"ן: קו העברת נתונים הוא קו פרטי של הלקוח, אשר מקשר בין שני אתרים. חברת בזק תחבר כל שתי נקודות בארץ או בעולם (לא כל מקום בעולם) בקו העברת נתונים בנל"ן לפי דרישות הלקוח. הלקוח יכול לקבוע קצב העברת נתונים ורמת אמינות של קו התקשורת (עד גבול מסוים ומחיר בהתאם לדרישות). קו נל"ן יקר ואינו מומלץ לאנשים פרטים, אלא לעסקים. קו העברת הנתונים הוא קו פיסי כלומר, קיים ערוץ תקשורת במסלול קבוע וידוע מראש המוקצב אך ורק להעברת הנתונים.

קו סיפרנת: קו זה דומה לקו קה"ן. כלומר, הוא מקיים קשר בין שתי נקודות. אבל בניגוד לקו העברת נתונים, הקו בין שתי הנקודות איננו פיסי, אלא לוגי. כלומר, אם קיים קה"ן בין ירושלים לתל אביב וטרקטור חתך אותו, ערוץ התקשורת ישאר מנותק עד שהקה"ן יחובר פיסית. לעומת זאת, בסיפרנת החיבור הפיסי בין הנקודה בירושלים לנקודה בתל אביב אינו קבוע. אם נחתך הקו בין ירושלים לתל אביב, הרשת תיישם את ערוץ התקשורת הלוגי בדרך חלופית, למשל דרך חיפה, ללא תלות בתיקון הקו בין ירושלים לתל אביב.

ISDN: זהו התקן לרשת התקשורת העתידית. אין עדיין מקום בעולם שבו קיימת ופועלת רשת ISDN מלאה, אך יש ניסיונות רבים בתחום זה. גם בישראל החלו מספר ניסיונות בתקן ISDN. בזק אמורה להתחיל בשיווק של רשת ISDN לצרכנים בשנת 1995. שירותי תקשורת אסינכרונית ינתנו על ידי רשת ISDN בדומה לרשת ישראנת.



נספחים

1. תקציר פקודות של מערכת ההפעלה DOS גירסה 5
2. טבלאות ASCII וקוד סריקה במקלדת
3. פסיקות עיקריות בחומרה ובתוכנה
4. כתובות מוחלטות ב-RAM לשימוש BIOS
5. תכניות דוגמה



נספח 1

תקציר פקודות של מערכת ההפעלה DOS גירסה 5

מקרא:

* פנימית/חיצונית: מיקום הפקודה במערכת ההפעלה DOS. פקודה פנימית נמצאת במעבד הפקודות וטעונה דרך קבע בזיכרון (TSR). פקודה חיצונית היא קובץ תכנית חיצונית של מערכת ההפעלה.

* שימוש ב- *?: מציין אם ניתן להשתמש בתווי פרא אלה על מנת להגדיר שם, או שמות, של הקבצים בפקודה (WildCards).

* אופציות שונות: פירוט האפשרויות השונות להפעלת הפקודה או התכנית.

* בכל פקודות DOS ניתן להשתמש באופציות הבאות המציינות מיעון שונה לקלט/פלט:

< File-Name	משמעות הסימון שהקלט לתכנית מתוך File-Name ולא מהמקלדת המהווה ברירת מחדל לקלט.
> File-Name	משמעות הסימון שפלט התכנית יירשם לקובץ File-Name ולא למסך המהווה ברירת מחדל לפלט.
| DOS-Command	משמעות הסימון שהפלט מהתכנית לא ייכתב למסך, אלא ישמש כקלט לתכנית DOS-Command.

את פקודות מערכת ההפעלה DOS ערכנו במספר סעיפי משנה, לפי מיון של שם הפקודה (באנגלית):

1. **פקודות לטיפול בקבצים**
2. **פקודות לטיפול בספריות**
3. **פקודות לטיפול במערכת הדיסקים**
4. **פקודות כלליות של מערכת ההפעלה**
5. **פקודות של קובץ CONFIG.SYS**
6. **תכניות הפעלה להתקנים (Device Drivers)**

ראה את הרשימה הביבליוגרפית הכוללת את הספרות של חברת מיקרוסופט (MS) וספרות אחרת.

פקודות לטיפול בקבצים

הפקודה	תבנית הפקודה	פנימית/ חיצונית	שימוש ב- *?	אופציות עיקריות
שינוי תכונה לקובץ	ATTRIB ±Attribute File-Name	חיצונית	כן	±A הרם/הורד את סיבית Archive ±H הסתר/בטל הסתרה של קבצים נסתרים (Hidden) ±R הרם/הורד סיבית הקריאה בלבד (ReadOnly) ±S הרם את סיבית המערכת (System)
העתקת קבצים	COPY Source-File Dest-File	פנימית	כן	/A קבצי ASCII העתק עד לסוף קובץ (^Z) /B קבצי שפת מכונה. העתק את הקובץ לפי גודלו + פעולת APPEND. הקבצים יועתקו בסוף קובץ היעד /V בדוק שהקבצים אכן הועתקו כראוי
השוואת קבצים	COMP File-Name1 File-Name2	חיצונית	כן	/A הצג הבדלים בצורה של תווי ASCII /C אין הבדל בין אותיות קטנות/גדולות /D הצג הבדלים בצורה עשרונית (Decimal) /L הצגת גם את מספר השורה במקרה של הבדלים /N=n השווה רק את n השורות הראשונות בקבצים
מחיקת קבצים	DEL File-Name	פנימית	כן	/P בקש אישור לפני מחיקת קובץ
רשימת קבצים הצגה של גודל הקבצים המוצגים והמקום הפנוי שנותר בכונן	DIR File-Name	פנימית	כן	/Aa הצג את הקבצים שהתכונה שלהם היא a. התכונה a מציינת: A עבור קבצים שסיבית Archive בהם מורמת, D עבור ספריות, H עבור קבצים נסתרים, R עבור קבצים לקריאה בלבד, S עבור קובצי System. - מציין רק קבצים שבהם התכונה המצויינת אינה קיימת /B רשימת קבצים מקוצרת ללא כותרות /L הצגת שמות הקבצים באותיות קטנות באנגלית (Lower Case)

פקודות לטיפול בקבצים (המשך)

הפקודה	תבנית הפקודה	פנימית/ חיצונית	שימוש ב- *?	אופציות עיקריות
				/Oo הצג את שמות הקבצים על פי הסדר o. התו o מציין מיון על פי: D תאריך ושעה, G מיון על פי סיומת הקובץ, G מיון על פי ספריות ראשונות, N מיון על פי שם קובץ, S מיון על פי גודל. - מציין מיון בסדר הפוך: מגדול לקטן /P הצג עמוד אחר עמוד וחכה למקש כלשהו להמשך /S הצג שמות קבצים גם בתת הספריות (במקרה של \ יתבצע חיפוש בכל הדיסק) /W הצגת רחבה של שמות הקבצים בלבד
פתיחה מהירה של קבצים	FASTOPEN Drive:=Number	חיצונית	לא	/X נתוני הקבצים לפתיחה מהירה יישמרו בזיכרון Expanded ולא יתפסו מקום מתחת ל-640KByte Number - מספר הקבצים המירבי שיש לשמור
השוואת קבצים	FC File-Name1 File-Name2	חיצונית	כן	/A התייחסות והשוואה כקבצי טקסט (ASCII) /B התייחסות והשוואה כקבצים בינאריים /C אין התייחסות להבדלים בין אותיות קטנות לגדולות באנגלית (Lower & Upper) /L השוואה כקובצי טקסט. דיווח מלא על הבדלים /N הצג מספרי שורות בהשוואת קובצי טקסט /T התייחסות לרווחים וטבולטורים כתווים נפרדים. התייחסות ברירת מחדל מרחיבה רווחים לטבולטור כשצריך השוואה מסוג זה /W צמצום ודחיסה של רווחים וטבולטורים לרווח יחיד ואז ביצוע של ההשוואה
הדפסת קבצים	PRINT File-Name	חיצונית	כן	/T מחק מהתור את כל הקבצים המחכים להדפסה /C מחק מהתור קבצים מסוימים (אפשרויות נוספות בפרק 4 סעיף 4.3.6)

פקודות לטיפול בקבצים (המשך)

הפקודה	תבנית הפקודה	פנימית/ חיצונית	שימוש ב- ?*	אופציות עיקריות
שינוי שם של קובץ	REN Old-File New-File	פנימית	כן	
הצגת תוכן קובץ	TYPE File-Name	פנימית	לא	
החלפה / עדכון קבצים	REPLACE Source-File Dest-File	חיצונית	כן	A/ מעתיק קבצים חדשים ולא מחליף קיימים. לא אפשרי יחד עם U, S P/ מבקש אישור לפני כל החלפה / הוספה R/ מחליף / מוסיף גם קובצי קריאה בלבד ReadOnly S/ מחפש גם בתת ספריות אם קיים קובץ מתאים U/ מחליף רק קבצים אשר היעד (Dest) מעודכן פחות מאשר המקור (בדיקת תאריך ושעה) W/ מבקש אישור העתקה כללי לפני תחילת ביצוע
הכנס מנגנון נעילה ושיתוף קבצים. לשימוש בעיקר ברשת תקשורת	SHARE	חיצונית	לא	F:Size/ כמה מקום לשמור עבור מידע לשיתוף ונעילה של הקבצים השונים (ברירת מחדל - 2048 בתים) L:Number/ מספר הנעילות המותרות במערכת
העתקה סלקטיבית	XCOPY Source-File Dest-File	חיצונית	כן	A/ העתק רק קבצים שסיבית Archive שלהם במצב + D:DATE/ העתק רק קבצים ששונו לאחר או במשך DATE E/ העתק גם ספריות ריקות M/ העתק קבצים שסיבית Archive במצב + , לאחר העתקה שנה מצב סיבית Archive ל - P/ בקש אישור לפני העתקה של כל קובץ S/ העתק גם תת ספריות ולא רק קבצים V/ בדוק כל קובץ מועתק וודא נכונות כתיבה W/ מבקש אישור כללי לפני תחילת העתקה

פקודות לטיפול בקבצים (המשך)

הפקודה	תבנית הפקודה	פנימית/ חיצונית	שימוש ב- *?	אופציות עיקריות
ביטול מחיקה של קבצים (שחזור ממחיקה)	UNDELETE File-Name	חיצונית	כן	ALL/ בטל מחיקה של כל הקבצים שניתן לשחזר DOS/ שימוש בספרית MS-DOS בלבד DT/ שימוש בקובץ מעקב אחר מחיקות בלבד לשחזור LIST/ הצג את כל הקבצים שניתן לשחזר

פקודות לטיפול בספריות של מערכת ההפעלה

הפקודה	תבנית הפקודה	פנימית/ חיצונית	שימוש ב- *?	אופציות עיקריות
הגדרת מסלול חיפוש של קובצי נתונים	APPEND Path-Name1;...	חיצונית	לא	E/ מאפשר לשמור את הספריות המוגדרות במסלול החיפוש בסביבת DOS לשם חיפוש מהיר (אפשרי רק בהפעלה ראשונה) PATH:ON/ חפש במסלול החיפוש גם קבצים שמוגדר עבורם מסלול חיפוש מלא בפקודה, או בתכנית המבקשים לפתוח אותם PATH:OFF/ חפש רק קבצים שלא מוגדר עבורם מסלול חיפוש מלא, אלא רק שמם X:ON/ מאפשר ליותר תכניות להשתמש במסלול החיפוש X:OFF/ מאפשר לפחות תכניות להשתמש במסלול חיפוש (תלוי בצורת השימוש של התכנית ב-DOS)
שינוי תת-ספריה	CD Path-Name	פנימית	לא	
יצירת תת-ספריה	MD Path-Name	פנימית	לא	
הגדרת מסלול חיפוש לקובצי תכניות	PATH Path-Name1;Path-Name2..	פנימית	לא	
מחיקת תת-ספריה	RD Path-Name	פנימית	לא	
הצגת עץ מבנה של ספריות	TREE Drive:	חיצונית	לא	A/ שימוש בסימני ASCII ולא בגרפיקה בציור העץ F/ הצג את שמות הקבצים בכל תת-ספריה שמוצגת

פקודות לטיפול במערכת הדיסקים

הפקודה	תבנית הפקודה	פנימית/ חיצונית	שימוש ב- *?	אופציות עיקריות
שינוי שם לכונן (Redirection)	ASSIGN Drive: Drive:	חיצונית	לא	/STATUS הצג אם קיימת הפניה לכונן
גיבוי דיסק	BACKUP File-Names Drive:	חיצונית	כן	/A העתק בסוף הנתונים בדיסקט. פעולת Append /D:Date העתק רק קבצים ששונו בזמן או לאחר Date /F:Size בצע FORMAT לדיסקט שאינו מוכן לפעולה לגודל המוגדר ב-Size /L:File-Name רשום לקובץ File-Name את תהליך הגיבוי. בקרה למשתמש /M גיבוי של קבצים שסיבית Archive שלהם היא + בגמר העתקה הורד סיבית זאת ל - /S בצע גיבוי גם של תת הספריות /T:Time גיבוי קבצים שנוצרו בזמן או לאחר Time
בדיקת דיסק/דיסקט	CHKDSK Drive:	חיצונית	לא	/V הצג רשימת קבצים וספריות בזמן בדיקה /F אפשר תיקון של איזורים פגועים File-Name בדיקה האם הקובץ רציף או אינו רציף (Fragmented)
השוואת דיסקטים	DISKCOMP Drive: Drive:	חיצונית	לא	/1 השווה רק צד אחד /8 השווה רק 8 גזרות במסלול
העתקת דיסקטים	DISKCOPY Drive: Drive:	חיצונית	לא	/1 העתק רק צד אחד /V בגמר העתקה יש לוודא שהנתונים הועתקו נכון
עריכת דיסק/דיסקט לשימוש	FORMAT Drive:	חיצונית	לא	/B השאר מקום לגרעין מערכת הפעלה בלא להכניסם /1 בצע הכנה על צד אחד בלבד /4 הכן דיסקט 360KB על כונן 1.2MB

פקודות לטיפול במערכת הדיסקים (המשך)

הפקודה	תבנית הפקודה	פנימית/ חיצונית	שימוש ב- *?	אופציות עיקריות
				/F:Size הגדר קיבולת דיסקט:1.44, 720, 1.2, 360 /N:Sectors הגדר מספר גזרות במסלול /Q פורמט מהיר, שימוש בפורמט קיים ושמירת מבנה הדיסק לצורך שחזור עתידי של הדיסק/דיסקט /S הכנס גם קובצי גרעין מערכת הפעלה /T:Trackes הגדר מספר מסלולים בדיסקט /V:Name בגמר התהליך יתן לדיסקט את השם Name
צירוף של כונן למסלול חיפוש בכונן אחר	JOIN Drive: Drive:\Path	חיצונית	לא	Drive: הכונן שיש לצרף. Drive:\Path מסלול החיפוש שאליו יצורף הכונן
שמירת מידע חיוני של דיסק לשם שחזור במקרה שבו נגרם נזק לדיסק	MIRROR Drive:	חיצונית	לא	/1 שמירה של הנתונים החדשים בלבד entries- מספר הקבצים המירבי למעקב /PARTN שמירת טבלת המחיצות של דיסק ע"ג דיסקט /Tdrive טוען את התכנית העוקבת אחר מחיקת קבצים עבור הכונן Drive /U מוריד מזיכרון את התכנית למעקב אחר מחיקה
שינוי שם לכונן / ספריה (הגדרת כונן בפועל)	SUBST Drive-Letter Drive-Path	חיצונית	לא	/D בטל את שם הכונן בפועל Drive-Letter (Virtual Drive)
העתק גרעין מערכת הפעלה לדיסק או דיסקט	Sys Src-Drive Dest-Drive	חיצונית	לא	Src-Drive כונן וספריה המכילים קובצי מערכת (לא חייבים להגדיר) Dest-Drive הכונן שאליו יש להעתיק את הקבצים

פקודות לטיפול במערכת הדיסקים (המשך)

הפקודה	תבנית הפקודה	פנימית/ חיצונית	שימוש ב- *?	אופציות עיקריות
שחזור לאחר ביצוע FORMAT לדיסק/דיסקט	UNFORMAT Drive:	חיצונית	לא	J/ בדיקת התאמה בין קבצים שנוצרו מתכנית MIRROR לבין נתוני הדיסק/דיסקט L/ הצגה של כל הקבצים והספריות הכלולים בטבלת המחיצות שנשמרה בפקודת MIRROR והצגת טבלת המחיצות. P/ כל הודעות השגיאה לסוגיהן יודפסו למדפסת PARTN/ שחזור של טבלת המחיצות (Partition) TEST/ בדיקת אפשרות השחזור מבלי לשחזר U/ היחלצות מפורמט מוטעה של דיסק/דיסקט (יעשה שחזור של הדיסק) מבלי להשתמש במידע שנשמר בפקודת MIRROR
שחזור קבצים מקובץ גיבוי (BACKUP)	RESTORE Drive: File-Names	חיצונית	כן	A:Date/ שחזור קבצים שנוצרו או עודכנו בתאריך או לאחר תאריך Date B:Date/ שחזור קבצים שנוצרו או עודכנו בתאריך או לפני תאריך Date E:Time/ שחזור קבצים שנוצרו או עודכנו בזמן או לפני זמן Time L:Time/ שחזור קבצים שנוצרו או עודכנו בזמן או אחרי זמן Time M/ שחזור של קבצים ששונו לאחר גיבוי אחרון N/ שחזור של קבצים שלא קיימים על דיסק היעד P/ בקשת אישור לפני כל אחזור של קובץ

פקודות כלליות של מערכת ההפעלה

הפקודה	תבנית הפקודה	פנימית/ חיצונית	שימוש ב- *?	אופציות עיקריות
שינוי תאריך	DATE	פנימית	לא	
הפעלת תכנית דפדוף בפקודות DOS קודמות	DOSKEY	חיצונית	לא	/BUFSIZE=N כאשר N מספר הפקודות המירבי לאחור שהתכנית תשמור /H הצגה של הפקודות האחרונות שניתנו ל-DOS /REINSTALL הכנסת עותק חדש של תכנית DOSKEY לזיכרון במקום עותק ישן /MACROS הצגת כל המקרו שהוגדרו בתכנית DOSKEY
הפעלת DOS Shell	DOSSHELL	חיצונית	לא	/B הצגה בשחור ולבן /G הצג בצורה גרפית (Graphics Mode) /T הצג בצורת טקסט (Text Mode)
הפעלת עורך מסך (Full Screen)	EDIT File-Name	חיצונית	לא	
הפעלת עורך שורה	EDLIN File-Name	חיצונית	לא	
הרחבת קובץ דחוס של תוכנת DOS-5	EXPAND File-name Dest	חיצונית	לא	File-Name שם הקובץ שיש להרחיב Dest שם הקובץ/קבצים והכונן לשם יירשמו נתוני הקבצים המורחבים
מצא בקובץ את מחרוזת החיפוש	FIND "String" File-Name	חיצונית	לא	/C מספר השורות שבהן מופיעה מחרוזת החיפוש /I אין התייחסות לאותיות קטנות/גדולות בחיפוש /N הצג את מספר השורה שבה מופיעה המחרוזת בכל קובץ החיפוש /V הצג את כל השורות שבהן מחרוזת החיפוש String איננה מופיעה

פקודות כלליות של מערכת ההפעלה (המשך)

הפקודה	תבנית הפקודה	פנימית/ חיצונית	שימוש ב- *?	אופציות עיקריות
טעינת טבלה גרפית	GRAFTABL	חיצונית	לא	
טעינת תכנית הדפסה של מסך גרפי	GRAPHICS	חיצונית	לא	
שנה שם דיסק/דיסקט	LABEL Drive:	חיצונית	לא	
הצג תפוסת זיכרון	MEM	חיצונית	לא	C/ הצגת נתוני תכניות, נתוני מצב, גודל תכניות מקום פנוי בזיכרון על פי סוג השימוש בזיכרון D/ הצג תכניות, Device Drivers ונתוני תוכנה שונים של מערכת ההפעלה DOS P/ הצג מצב תכניות הטעונות בזיכרון
כיוון פרמטרים של כרטיס טורי	MODE COMn: B,P,D,S,R	חיצונית	לא	n מספר הכרטיס הטורי (1 או 2) B מהירות העבודה (Baud Rate) P סוג סיבית הזוגיות (O,E,N) D מספר סיביות המידע (7 או 8) S מספר סיביות הסיום (1 או 2)
כרטיס מקבילי	MODE LPTn: c,l	חיצונית	לא	n מספר הכרטיס המקבילי (1 או 2) c מספר התוים ברוחב הדף (80 או 132)
מקלדת (מהירות)	MODE CON: RATE=r DELAY=d	חיצונית	לא	r קצב השידור של תווים רצופים למחשב (1 עד 32) d השהיה עד לזיהוי לחיצה רצופה (1 עד 4)

פקודות כלליות של מערכת ההפעלה (המשך)

הפקודה	תבנית הפקודה	פנימית/ חיצונית	שימוש ב- *?	אופציות עיקריות
מסך	MODE COn MODE MONO או MODE BWn	חיצונית	לא	n אופן הפעולה של המסך (80 או 40) MONO מסך מונוכרום CO הצגה בצבעים BW הצגה בשחור/לבן
הצג נתונים בדפים	MORE < File-Name	חיצונית	לא	l מספר שורות באינץ (6 או 8)
שינוי הסימן המנחה	PROMPT New-Prompt	פנימית	לא	$B הצג את הסימן | $D הצג את התאריך $E הצג את התו Escape $G הצג את התו > $H הצג את הסימן BackSpace $L הצג את הסימן < $N הצג את שם כונן ברירת המחדל $P הצג את הספריה והכונן (ברירת המחדל) $Q הצג את הסימן = $T הצג את השעה $$ הצג את הסימן $ $_ הצג את הסימן LineFeed ו-Carriage Return. New-Prompt - אוסף תווי ASCII שיופיעו על המסך
מיין נתונים	SORT < File-Name או | SORT	חיצונית	לא	/R מיון השורות בסדר הפוך /+N החל מיון מעמודה N בכל שורה
שינוי זמן מערכת	TIME	פנימית	לא	
הצג גירסת DOS	VER	פנימית	לא	

פקודות כלליות של מערכת ההפעלה (המשך)

הפקודה	תבנית הפקודה	פנימית/ חיצונית	שימוש ב- *?	אופציות עיקריות
שנה דגל בדיקת אמינות כתיבה לדיסק/דיסקט	VERIFY	פנימית	לא	ON - הרם דגל בדיקה. בכל כתיבה לדיסק מתבצעת קריאה חזרה על מנת לבדוק את תקינות הכתיבה OFF - הורד דגל בדיקה
הצג שם דיסק/דיסקט	VOL Drive:	פנימית	לא	
קביעת מספר המהדורה ש-DOS-5 תדווח לקובץ התכנית	SETVER File-Name Ver-Num SETVER File-Name/D	חיצונית	לא	File-Name - שם קובץ (אפשר תווי פרא) שאליו מדווחים Ver-Number - מספר הגירסה ש-DOS תדווח לקובץ D/ מחק את הקובץ מרשימת ה-SETVER

פקודות של קובץ CONFIG.SYS

קובץ CONFIG.SYS מכיל פקודות לשינוי תצורת החומרה וסביבת מערכת ההפעלה של המחשב. קובץ זה מתבצע בתהליך BOOT על ידי תכנית האתחול של מערכת ההפעלה DOS. הפקודות בקובץ CONFIG.SYS מאפשרות לשפר את ביצועי מערכת המחשב.
פקודות אלו אינן ניתנות להרצה ברמת הסימן המנחה (Prompt) של DOS, אלא רק מתוך קובץ CONFIG.SYS ורק בתהליך BOOT.

הפקודה	תבנית הפקודה והסבר	אופציות עיקריות
BREAK	BREAK=ON/OFF שינוי אופן ההתייחסות של מערכת ההפעלה להפרעת משתמש לתכנית באמצעות לחיצה על Ctrl+C (^C).	OFF - התייחסות ללחיצה על צירוף Ctrl+C תעשה רק בקריאה מהמקלדת או כתיבה למסך ON - התייחסות ללחיצה על הצירוף Ctrl+C בכל פעולות קלט/פלט במהלך ביצוע התכנית
BUFFERS	BUFFERS=Num מספר בלוקים בזיכרון שישמשו את מערכת ההפעלה לשמירת נתונים לפעילות שוטפת.	Num - מספר הבלוקים (עד 99) כאשר משתמשים בזיכרון בסיסי עד 640KB, או עד 10000 אם משתמשים בזיכרון Expanded /X - התכנית תשתמש בזיכרון Expanded
COUNTRY	COUNTRY=Code,Cp,File-Name שינוי צורת הרישום של תאריך, שעה, מפרידים עשרוניים וכו', על מנת להתאימם לארץ שבה פועלים.	Code - מספר הארץ על פי הקידומת הבינלאומית. לדוגמה, ישראל 972, ארה"ב 001 וכו' Cp - מספר גופן (Code Page) File-Name - הקובץ המכיל את המידע COUNTRY.SYS
DEVICE	DEVICE=Device-Name הטענת תכנית הפעלה להתקנים (Device Driver).	Device-Name - שם התכנית שיש לטעון לזיכרון
DRIVPARM	DRIVPARM= הגדרת כונן דיסקטים שלא מוכר ל-BIOS.	/D:Disk - מספר כונן (A=0, B=1 וכו') /F:Type - סוג כונן (720, 360 וכו'). הגדרה של סוג לא תבוא עם הגדרת T, S, H

פקודות של קובץ CONFIG.SYS (המשך)

הפקודה	תבנית הפקודה והסבר	אופציות עיקריות
		/H:Head - מספר ראשים קוראים /S:Sector - מספר גזרות במסלול /T:Track - מספר מסלולים במשטח /C - הכונן מספק חיווי חומרה על פתיחת הדלת
FCBS	FCBS=NOpen,NClose הגדרת מספר מירבי של טבלאות בקרה לקבצים (File Control Blocks).	NOpen - מספר FCB שניתן לפתוח בבת-אחת NClose - מספר קבצים המכילים FCB שמערכת ההפעלה אינה יכולה לסגור אוטומטית
FILES	FILES=Num הגדרת מספר מירבי של קבצים פתוחים.	Num - מספר הקבצים הפתוחים. עד 255
INSTALL	INSTALL=File-Name הרצת תכנית באמצעות פקודה בקובץ CONFIG.SYS.	File-Name - שם תכנית חוקי: FASTOPEN ,KEYB NLSFUNC ,SHARE
LASTDRIVE	LASTDRIVE=Drive-Letter קביעת שם הכונן המירבי האפשרי במערכת.	Drive-Letter - האות A-Z
REM	הערה בקובץ. אין התייחסות לתוכן השורה.	
STACKS	STACKS=Num,Size קביעה של גודל המחסנית שתעמוד לרשות מערכת ההפעלה.	Num - מספר המחסניות. ערך מ-0 עד 64 Size - גודל כל מחסנית. ערך מ-0 עד 512 כאשר הערך של Num ו-Size הוא 0, מערכת ההפעלה תשתמש במחסנית המוגדרת על ידי המשתמש

תכניות הפעלה להתקנים (Device Drivers)

תכניות הפעלה להתקנים (Device Drivers) מאפשרות להפעיל התקני חומרה לא תקניים. בכך הן מאפשרות למערכת ההפעלה DOS להפעיל התקנים שאיננה מכירה ולהרחיב את אפשרויות הפעולה של מערכת המחשב. תכניות הפעלה אלו נטענות לזיכרון המחשב בשלב BOOT של המערכת. את הפקודות לטעינת Device Drivers כותבים בקובץ CONFIG.SYS.

התכנית (Device Driver)	תבנית הפקודה והסבר	אופציות עיקריות
ANSI.SYS	DEVICE=ANSI.SYS תכנית ANSI.SYS מאפשרת לתכנית יישום לשלוח פקודות הצגה למסך לפי תקן ANSI. תקן ANSI משמש בעיקר במסופים המחוברים למחשב מרכזי.	/X - בלוח מקשים מורחב, שימוש באופציה זו מקנה משמעות שונה למקשים המשמשים לאותו תפקיד אך ממוקמים במקום שונה. למשל מקש PgUP ומקש Page Up
DISPLAY.SYS	DEVICE=DISPLAY.SYS CON:=(Scr-Type,Code,N) תכנית DISPLAY.SYS מאפשרת להשתמש במנגנון החלפת גופנים הקיים במתאמי מסך EGA ומעלה.	Scr-Type - סוג מסך VGA, EGA וכו' Code - סוג הגופן (Code Page) N - מספר גופנים (Code Pages)
DRIVER.SYS	DEVICE=DRIVER.SYS הגדרת כונן דיסקטים שלא מוכר ל-BIOS.	/D:Disk - מספר כונן (A=0, B=1 וכו') /F:Type - סוג כונן (720, 360 וכו'). הגדרה של סוג לא תבוא עם הגדרת T, S, H /H:Head - מספר ראשים קוראים /S:Sector - מספר גזרות במסלול /T:Track - מספר מסלולים במשטח /C - הכונן מספק חיווי חומרה לפתיחה של הדלת
EMM386.SYS XMAEM.SYS QEMM.SYS	DEVICE=EMM386.SYS תכנית להגדרת זיכרון מסוג Extended כזיכרון מסוג Expanded במחשבי 386.	

תכניות הפעלה להתקנים (Device Drivers) (המשך)

התכנית (Device Driver)	תבנית הפקודה והסבר	אופציות עיקריות
HIMEM.SYS	DEVICE=HIMEM.SYS תכנית להגדרת זיכרון מסוג Extended כזיכרון מסוג Expanded. הפקודה שימושית כאשר תכנית היישום יכולה לפעול רק בזיכרון Expanded.	
PRINTER.SYS	DEVICE=PRINTER.SYS LPTn=(Type,Code,N) החלפת גופנים (Code Pages) במדפסת. לא כל המדפסות תומכות בחילוף גופנים.	Type - סוג מדפסת (5202, 4208, 4201) Code - סוג הגופן (Code Page) N - מספר גופנים
RAMDRIVE.SYS VDISK.SYS	DEVICE=RAMDRIVE.SYS Disk Sector Entries התקנת דיסק בפועל (Virtual Disk).	Disk - גודל זיכרון RAM ב-KB שיוקצה לדיסק Sector - גודל סקטור בדיסק (בתים) Entries - מספר כניסות בספריית השורש /E - הדיסק בפועל ישתמש בזיכרון Extended /A - הדיסק בפועל ישתמש בזיכרון Expanded
SMARTDRV.SYS	DEVICE=SMARTDRV.SYS Size התקנת תכנית זיכרון מטמון לדיסק (Disk Cache).	Size - גודל הזיכרון ב-KB שיוקצה לדיסק /A - הדיסק ישתמש בזיכרון Expanded ולא בזיכרון Extended
XMA2EMS.SYS	DEVICE=XMA2EMS.SYS תכנית לניהול זיכרון מסוג Expanded.	

נספח 2

טבלאות ASCII וקוד סריקה במקלדת

תווי בקרה בטבלת ASCII

ערך DEC	ערך HEX	Ctrl	קיצור	משמעות	ASCII
00	00		Nul	Null תו ריק	
01	01	^A	SOH	Start of Header תחילת ראש התקשורת	☺
02	02	^B	STX	Start of Text תחילת טכסט	☻
03	03	^C	ETX	End of text סיום טכסט	♥
04	04	^D	EOT	End of trans סיום תקשורת	♦
05	05	^E	ENQ	Enquiry מה המצב בצד השני של הקו	♣
06	06	^F	ACK	Positive Acknowledge התקבלו נתונים תקינים	♠
07	07	^G	BEL	Bell הפעל פעמון (צפצוף למפעיל)	•
08	08	^H	BS	Back Space הזז ראש כותב צעד אחד אחורה	◘
09	09	^I	HT	Horizontal Tab הקפץ ראש כותב אופקית קדימה	○
10	A	^J	LF NL	Line Feed or New Line התקדם שורה	◙
11	B	^K	VT	Vertical Tab קפוץ קדימה מספר שורות	♂
12	C	^L	FF	Form Feed קפוץ דף שלם קדימה	♀
13	D	^M	CR	Carriage Return החזר ראש כותב לתחילת שורה	♪
14	E	^N	SO	Shift Out צא ממצב תקשורת	♫
15	F	^O	SI	Shift In חזור למצב תקשורת	☼
16	10	^P	DLE	Data Link Escape צא מהעברת נתונים בינארית	►
17	11	^Q	Xon DC1	Device Control 1 המשך בשליחת נתונים	◄
18	12	^R	DC2	Device Control 2 שליטה בהתקן מס' 2	↕
19	13	^S	Xoff DC3	Device Control 3 עצור שליחת נתונים	‼
20	14	^T	DC4	Device Control 4 שליטה בהתקן מס' 4	¶
21	15	^U	NAK	Negative Acknowladge התקבלו נתונים שגויים	§
22	16	^V	SYN	Synchronous Idle סינכרון בין צדדים ללא תקשורת	▬
23	17	^W	ETB	End of Trans Block סיום העברת בלוק נתונים	↨
24	18	^X	CAN	Cancel בטל הודעה אחרונה	↑
25	19	^Y	EM	End of Medium סיום עבודה בערוץ תקשורת זה	↓
26	1A	^Z	ETF SUB	Substiute עבור לערוץ תקשורת חלופי End of Text File סיום קובץ טכסט	→
27	1B		ESC	Escape חזור לרמה קודמת בסביבת העבודה שלך	
28	1C		FS	File separator מפריד בין שני קבצים בתקשורת	
29	1D		GS	Group Separator מפריד בין קבוצות של קבצים	
30	1E		RS	Record Separator מפריד בין רשומות	
31	1F		US	Unit Separator מפריד בין יחידות	
32	20		SP	Space רווח של תו אחד	

המשך טבלת ASCII

32-	64- @	96-	128- א	160- á	192- └	224- α
33- !	65- A	97- a	129- ב	161- í	193- ┴	225- ß
34- "	66- B	98- b	130- ג	162- ó	194- ┬	226- Γ
35- #	67- C	99- c	131- ד	163- ú	195- ├	227- π
36- $	68- D	100- d	132- ה	164- ñ	196- ─	228- Σ
37- %	69- E	101- e	133- ו	165- Ñ	197- ┼	229- σ
38- &	70- F	102- f	134- ז	166- ª	198- ╞	230- µ
39- '	71- G	103- g	135- ח	167- º	199- ╟	231- τ
40- (	72- H	104- h	136- ט	168- ¿	200- ╚	232- Φ
41-)	73- I	105- i	137- י	169- ⌐	201- ╔	233- Θ
42- *	74- J	106- j	138- ך	170- ¬	202- ╩	234- Ω
43- +	75- K	107- k	139- כ	171- ½	203- ╦	235- δ
44- ,	76- L	108- l	140- ל	172- ¼	204- ╠	236- ∞
45- -	77- M	109- m	141- ם	173- ¡	205- ═	237- φ
46- .	78- N	110- n	142- מ	174- «	206- ╬	238- ε
47- /	79- O	111- o	143- ן	175- »	207- ╧	239- ∩
48- 0	80- P	112- p	144- נ	176- ░	208- ╨	240- ≡
49- 1	81- Q	113- q	145- ס	177- ▒	209- ╤	241- ±
50- 2	82- R	114- r	146- ע	178- ▓	210- ╥	242- ≥
51- 3	83- S	115- s	147- ף	179- │	211- ╙	243- ≤
52- 4	84- T	116- t	148- פ	180- ┤	212- ╘	244- ⌠
53- 5	85- U	117- u	149- ץ	181- ╡	213- ╒	245- ⌡
54- 6	86- V	118- v	150- צ	182- ╢	214- ╓	246- ÷
55- 7	87- W	119- w	151- ק	183- ╖	215- ╫	247- ≈
56- 8	88- X	120- x	152- ר	184- ╕	216- ╪	248- °
57- 9	89- Y	121- y	153- ש	185- ╣	217- ┘	249- ∙
58- :	90- Z	122- z	154- ת	186- ║	218- ┌	250- ·
59- ;	91- [	123- {	155- ø	187- ╗	219- █	251- √
60- <	92- \	124- ¦	156- £	188- ╝	220- ▄	252- ⁿ
61- =	93-]	125- }	157- †	189- ╜	221- ▌	253- ²
62- >	94- ^	126- ~	158- │	190- ╛	222- ▐	254- ■
63- ?	95- _	127- ⌂	159- —	191- ┐	223- ▀	255-

קוד סריקה וקוד ASCII במקלדת רגילה

המקש			לחיצה על מקש+Shift			CTRL לחיצה על מקש+Ctrl			לחיצה על מקש+Alt		
	סריקה	ASCII	משמע'	סר'	ASC	משמע'	סר'	ASC	משמע'	סר'	ASC
F1	59	0	F11	84	0	F21	94	0	F31	104	0
F2	60	0	F12	85	0	F22	95	0	F32	105	0
F3	61	0	F13	86	0	F23	96	0	F33	106	0
F4	62	0	F14	87	0	F24	97	0	F34	107	0
F5	63	0	F15	88	0	F25	98	0	F35	108	0
F6	64	0	F16	89	0	F26	99	0	F36	109	0
F7	65	0	F17	90	0	F27	100	0	F37	110	0
F8	66	0	F18	91	0	F28	101	0	F38	111	0
F9	67	0	F19	92	0	F29	102	0	F39	112	0
F10	68	0	F20	93	0	F30	103	0	F40	113	0
Home (מספרי)	71	0		71	55		119	0			
Up (מספרי)	72	0		72	56						
PgUp (מספרי)	73	0		73	57		132	0			
Left (מספרי)	75	0		75	52		115	0			
5 (מספרי)	76	0		76	53						
Right (מספרי)	77	0		77	54		116	0			
End (מספרי)	79	0		79	49		117	0			
Down (מספרי)	80	0		80	50		145	0			
PgDn (מספרי)	81	0		81	51		118	0			
Ins (מספרי)	82	0		82	48						
Del (מספרי)	83	0		83	46						
Home	71	0		71	0		119	0			
Up Arrow	72	0		72	0						
Page Up	73	0		73	0		132	0			
Left Arrow	75	0		75	0		115	0			
Right Arrow	77	0		77	0		116	0			
End	79	0		79	0		117	0			
Down Arrow	80	0		80	0						
Page Down	81	0		81	0		118	0			
Insert	82	0		82	0						
Delete	83	0		83	0						
Back Space	14	8		14	8		14	127			
1 עד 9 מרכזי	2-10	49-57								120-128	0
0 מרכזי	11	48								129	0

משמע' = משמעות; סר' = סריקה; ASC = ASCII

59 F1 | 60 F2 | 1 Esc | 2 1 | 3 2 | 4 3 | 5 4 | 6 5 | 7 6 | 8 7 | 9 8 | 10 9 | 11 0 | 12 - | 13 = | 14 ◄ | 69 Num Lock | 70 Scroll Lock

61 F3 | 62 F4 | 15 ⇆ | 16 Q | 17 W | 18 E | 19 R | 20 T | 21 Y | 22 U | 23 I | 24 O | 25 P | 26 [| 27] | 28 ↵ | 71 7 | 72 8 | 73 9 | 78 -

63 F5 | 64 F6 | 29 Ctrl | 30 A | 31 S | 32 D | 33 F | 34 G | 35 H | 36 J | 37 K | 38 L | 39 ; | 40 ' | 41 ` | 75 4 | 76 5 | 77 6 | 28 +

65 F7 | 66 F8 | 42 ⇧ | 43 \ | 44 Z | 45 X | 46 C | 47 V | 48 B | 49 N | 50 M | 51 , | 52 . | 53 / | 54 ⇧ | 55 PrtSc | 79 1 | 80 2 | 81 3

67 F9 | 68 F10 | 56 Alt | 57 | 58 Caps | 82 0 | 83

מערך המקשים במקלדת רגילה

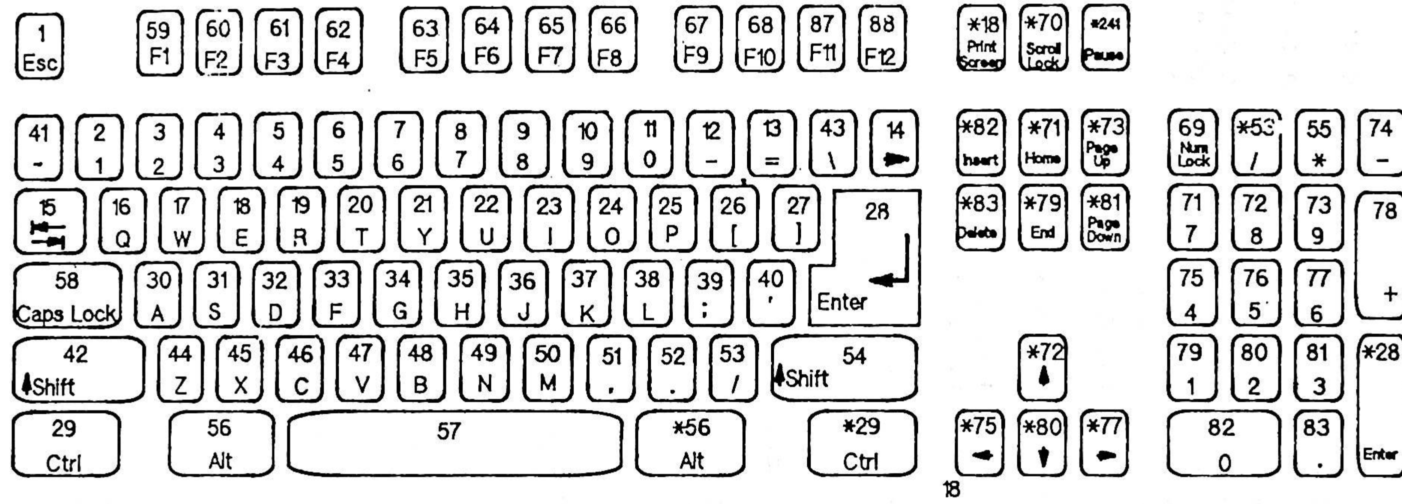
מקרא: הסימון התחתון על המקש – סימון המקש באנגלית (US).
הסימון העליון על המקש – קוד הסריקה של המקש.

קוד סריקה וקוד ASCII במקלדת מורחבת

(ראה הסבר על קריאה ממקלדת מורחבת – פונקציה 10Hex)

המקש			לחיצה על מקש+Shift			לחיצה על מקש+Ctrl			לחיצה על מקש+Alt		
	סריקה	ASCII	משמע'	סר'	ASC	משמע'	סר'	ASC	משמע'	סר'	ASC
F1	59	0	F11	84	0	F21	94	0	F31	104	0
F2	60	0	F12	85	0	F22	95	0	F32	105	0
F3	61	0	F13	86	0	F23	96	0	F33	106	0
F4	62	0	F14	87	0	F24	97	0	F34	107	0
F5	63	0	F15	88	0	F25	98	0	F35	108	0
F6	64	0	F16	89	0	F26	99	0	F36	109	0
F7	65	0	F17	90	0	F27	100	0	F37	110	0
F8	66	0	F18	91	0	F28	101	0	F38	111	0
F9	67	0	F19	92	0	F29	102	0	F39	112	0
F10	68	0	F20	93	0	F30	103	0	F40	113	0
F11	133	0		135	0		137	0		139	0
F12	134	0		136	0		138	0		140	0
Home (מספרי)	71	0		71	55		119	0			
Up (מספרי)	72	0		72	56		141	0			
PgUp (מספרי)	73	0		73	57		132	0			
- (לוח ימני)	74	45		74	45		142	0		74	0
Left (מספרי)	75	0		75	52		115	0			
5 (מספרי)	76	0		76	53		143	0			
Right (מספרי)	77	0		77	54		116	0			
+ (לוח ימני)	78	43		78	43		144	0		78	0
End (מספרי)	79	0		79	49		117	0			
Down (מספרי)	80	0		80	50		145	0			
PgDn (מספרי)	81	0		81	51		118	0			
Ins (מספרי)	82	0		82	48		146	0			
Del (מספרי)	83	0		83	46		147	0		163	0
Home	71	224		71	224		119	15		151	224
Up Arrow	72	224		72	224		141	224		152	224
Page Up	73	224		73	224		132	224		153	224
Left Arrow	75	224		75	224		115	224		155	224
Right Arrow	77	224		77	224		116	224		157	224
End	79	224		79	224		117	224		159	224
Down Arrow	80	224		80	224		145	224		160	224
Page Down	81	224		81	224		118	224		118	224
Insert	82	224		82	224		146	224		162	224
Delete	83	224		83	224		147	224		163	224
Back Space	14	8		14	8		14	127		14	0
Esc	1	27		1	27		1	27		1	27
1 עד 9 מרכזי	2-10	49-57								120-128	0
0 מרכזי	11	48								129	0
/ (לוח ימני)	224	47		224	47		149	0		164	0
* (לוח ימני)	55	42		55	42		150	0			

מקרא: משמע' = משמעות; סר' = סריקה; ASC = ASCII



מערך המקשים במקלדת מורחבת (101)

<u>מקרא:</u> הסימון התחתון על המקש – סימון המקש באנגלית (US).
הסימון העליון על המקש – קוד הסריקה של המקש.
*תח – לפני קוד הסריקה תח של המקש נשלח למחשב קוד סריקה 224.

נספח 3

פסיקות עיקריות בחומרה ובתוכנה

מספר פסיקה	תיאור הפסיקה והשימושים
00	חלוקה ב-0. פסיקה זו מתבצעת כאשר מחלקים ב-0.
01	צעד אחר צעד - Single Step. משמשת לתכניות Debuggers.
02	פסיקה שלא ניתנת למיסוך (NMI). משמשת במחשב לבקרת שגיאות Parity של הזיכרון הראשי במחשב.
03	נקודת שבירה - Break Point, משמשת תכניות Debuggers.
04	גלישה. מתבצעת כאשר בפעולה חשבונית יש גלישה.
05	הדפסת מסך - Print Screen. תוכן המסך מועבר למדפסת.
06	שמור לצרכים עתידיים.
07	שמור לצרכים עתידיים.
08	פסיקת חומרה של השעון Timer 0.
09	פסיקת חומרה של המקלדת - Keyboard.
0A	פסיקת חומרה ב-AT, שרשור לבקר פסיקות שני. ב-XT - פנוי.
0B	פסיקת חומרה לכרטיס תקשורת טורי שני COM2.
0C	פסיקת חומרה לכרטיס תקשורת טורי ראשון COM1.
0D	AT: פסיקת חומרה לכרטיס מקבילי שני LPT2; XT: דיסק קשיח
0E	פסיקת חומרה לכונן דיסקטים.
0F	פסיקת חומרה לכרטיס מקבילי ראשון LPT1.
10	Video BIOS, שירותי מסך.
11	BIOS, שירות לקבלת פירוט מרכיבי החומרה במחשב.
12	BIOS, שירות לקבלת פירוט של גודל זיכרון.
13	BIOS, שירותי גישה לכונני דיסקטים/דיסקים.
14	BIOS, שירותי תקשורת טורית ל- COM1 ,COM2.
15	BIOS, שירותי כונן קלטות (טייפ), זיכרון Extended.
16	BIOS, שירותי מקלדת.
17	BIOS, שירותי תקשורת מקבילית LPT1 ,LPT2.
18	BIOS, כניסה ל-Basic הצרוב ב-ROM על הלוח הראשי.
19	BIOS, שירות איתחול למחשב Bootstrep Loader.
1A	BIOS, שירותי זמן:קביעת שעון מערכת, שעון זמן אמת ואזעקה
1B	BIOS, פסיקה הנגרמת מלחיצה על צירוף המקשים Ctrl+Break.
1C	BIOS, פסיקת שעון המתבצעת 18.2 פעם בשניה.
1D	BIOS, איתחול מערכת הוידאו לאופני הפעולה שונים.
1E	BIOS, פרמטרים של מערכת כונני הדיסקטים.
1F	BIOS, איתחול טבלת התוים הגרפיים להצגה גרפית.

המשך טבלת הפסיקות

מספר פסיקה	תיאור הפסיקה והשימושים
20	DOS, סיום תוכנית Program Terminate.
21	DOS, שירותים כלליים של מערכת ההפעלה DOS.
22	DOS, כתובת סיום תוכנית. משמשת לחזרה למערכת ההפעלה.
23	DOS, כתובת יציאה לתכנית בלחיצה על Ctrl+Break.
24	DOS, תכנית לטיפול בשגיאות קריטיות.
25	DOS, שירות לקריאה ישירה מהדיסק/דיסקט.
26	DOS, שירות לכתיבה ישירה לדיסק/דיסקט.
27	DOS, שירות לסיום תכנית והשארתה בזיכרון כ-TSR.
28	DOS, תכנית שירות המציינת ש-DOS במצב מנוחה ומצפה לקלט.
29	DOS, כתיבה מהירה (FAST PUTCHAR).
2A	DOS, ממשק לרשת (Microsoft Network Interface).
2B-2D	DOS, שמור.
2E	DOS, טעינת תכניות לזיכרון תחת DOS.
2F	DOS, שירותים מורכבים (שרת הדפסות PRINT, ופרמטרים שונים של תכניות כמו SHARE, ASSIGN ועוד).
30-32	שמור.
33	פסיקת שירות לטיפול בעכבר.
34-3F	שמור.
40	BIOS, במחשבי AT משמש כהפניה לגישה לכונן דיסק/דיסקט.
41-46	פסיקות BIOS. פרמטרים לגישה לדיסק קשיח במחשבי AT ומעלה.
60-67	שמור לפסיקות משתמש.
50-57	פסיקות חומרה של בקר פסיקות שני במחשבי AT ומעלה.
5C	ממשק ל-BIOS של כרטיס תקשורת (Network BIOS).
67	זיכרון הרחבה Expanded. שירותי Device Driver של הזיכרון
70-77	פסיקות חומרה של בקר פסיקות שני במחשבי AT ומעלה.
80-85	שמור לשימוש BASIC.
86-EF	שמור לשימוש המפרש - BASIC Interpreter.
F0-FF	פנוי לתכניות משתמש.

נספח 4

כתובות מוחלטות ב-RAM לשימוש BIOS

הכתובת (HEX)	שימושים
00000-003FF	מרחב פסיקות. כל פסיקה 4 בתים, סה"כ 256.
00400-00407	כתובות של עד 4 מתאמי תקשורת טורית (COM).
00408-0040F	כתובות של עד 4 מתאמי תקשורת מקבילית (LPT).
00410-00411	פירוט של כל מרכיבי החומרה העיקריים במחשב.
00412	דגל איתחול.
00413-00414	גודל הזיכרון בפועל בכפולות של 1K.
00417-00418	דגלי מצב של המקשים המיוחדים במקלדת.
0041A-0041B	מצביע לראש התור של המקלדת.
0041C-0041D	מצביע לסוף התור של המקלדת.
0041E-0043D	תור למקשים שנלחצו במקלדת וטרם נקראו.
0043E-00448	פרמטרים לחיוויים מכונני הדיסקטים.
00449	אופן פעולה נוכחי של מתאם המסך.
0044A-0044B	מספר העמודות להצגה במסך.
0044C-0044D	כמות הבתים הנחוצה להצגת המסך האקטיבי (Page).
0044E-0044F	כתובת התחלתית במתאם המסך של הדף האקטיבי.
00450-00451	מיקום הסמן בדף 0 במסך.
00452-0045F	מיקום הסמן עבור דפים 1 עד 7.
00460-00461	אופן ההצגה של הסמן על גבי המסך.
00462	כמות דפי הטקסט שניתן להציג על גבי המסך.
00463-00464	כתובת התחלתית במרחב קלט/פלט של כרטיס מסך.
00466	ערך של טבלת הצבעים (Color Palette).
0046C-0046F	4 בתים שמחזיקים את ערך השעון של המערכת.
00470	דגל של שעון מערכת המציין שחלפה שעת חצות.

נספח 5

תכניות דוגמה

דיסקט תכניות הדוגמה כולל מספר סוגי קבצים:

README.DOC – הקובץ כולל רשימה של כל תכניות הדוגמה.

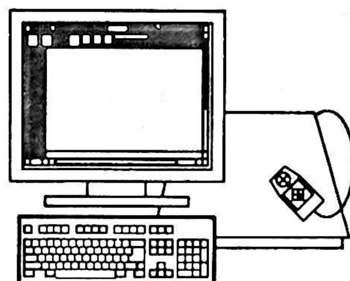
*.PAS – קובצי תכניות בשפת פסקל. תוכל ללמוד כיצד הן נכתבו וגם לערוך בהן שינויים כרצונך, כדי לנסות אפשרויות נוספות. התכניות כתובות בטורבו-פסקל והורצו במהדר Turbo-Pascal מגירסה 4.0 ומעלה. חלק מהתכניות חובה להריץ במהדר גירסה 6.0, או להתאים אותן להרצה במהדר בגירסה נמוכה יותר.

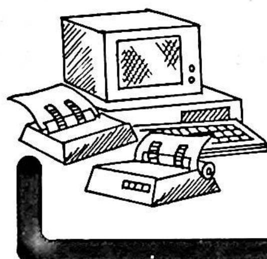
*.EXE – לכל התוכניות ניתנו גם קובצי ההרצה בשפת מכונה, בהתאמה לקובצי המקור. הן מיועדות לאלה שרק מעוניינים לראות את ההרצה עצמה, או שאין ברשותם מהדר פסקל מתאים. לצורך ההרצה נמצאים בדיסקט קבצים נוספים הנושאים סיומות כמו TXT, או BGI למשל.

מותר להעתיק את קובצי התכניות שבדיסקט.

אנו מקווים שתוכל להריץ את התכניות במחשב שברשותך ולהפיק מכך את מלוא התועלת שבלימוד בספר זה.

רשימת התכניות שלהלן וההסברים להן נלקטו מתוך קובצי התכניות שבדיסקט.





ASCII - הצגת תווי ASCII וערכם הדצימלי

תכנית זו משתמשת בטכניקה של כתיבה ישירה למסך, כדי להציג טבלה מלאה של הסימנים הגרפיים ב-ASCII. בכל דרך אחרת הסימנים המופיעים במקומות 7, 10, 13, לא היו מופיעים, אלא מתורגמים לפקודות.
הפעלת התכנית באמצעות הפקודה ASCII.
השגרה Writestrdirect כותבת מחרוזת תווים ישירות למסך, בדומה לשגרה המופיעה בתכנית SCRBDEMO.PAS.
ראה תכנית SCRDDEMO.PAS לשליטה על תכונות התו המוצג.
ראה נספח 2.

ATTRDEMO.PAS - שינוי תכונות קבצים

התכנית מאפשרת למשתמש לשנות תכונות (ReadOnly ,Hidden) של קבצים. היא גם מאפשרת לו לקבוע אילו קבצים ישונו, ומה תהיינה התכונות שלהם.
התכנית מציגה גם קבצים נסתרים וקובצי מערכת.
הפעלת התכנית על ידי הפקודה: ATTRDEMO <file-names>.
התכנית מציגה את גודל הקבצים, את התאריך שבו נוצרו ואת תכונותיהם.
לחיצה על מקש Y תתן לקובץ את התכונה המסומנת, לחיצה על מקש N תבטל את התכונה המסומנת מהקובץ, לחיצה על מקש Esc תסיים את התכנית.
ראה פרק 5 סעיף 5.4.3

BOOT.PAS - הצגת רשומת BOOT בדיסקט שבכונן A

התכנית ניגשת בצורה ישירה לדיסקט באמצעות BIOS, קוראת את רשומת BOOT מצד 0, מסלול 0, גזרה 1 בדיסקט ורושמת על גבי קובץ BOOT.TXT את תוכן הגזרה בהצגת Hex והצגת ASCII.
הפעלת התכנית באמצעות הפקודה BOOT.
ראה פרק 5 סעיפים 5.4.2.1,2

CLOCK.PAS - הצגת שעון בפינה ימנית של המסך

התכנית מציגה באופן קבוע שעון בפינה הימנית העליונה של המסך.
התכנית משתמשת בשעון התוכנה של המחשב האישי ונמצאת דרך קבע בזיכרון המחשב כתכנית TSR.
הפעלת התכנית באמצעות הפקודה CLOCK.

KBSCAN.PAS ,EKBSCAN.PAS

תכניות להצגת קוד הסריקה וקוד ASCII של מקשים במקלדת.
תכנית עבור לוח מקשים רגיל: KBSCAN.PAS
תכנית עבור לוח מקשים מורחב EKBSCAN.PAS

הפעלת התכנית באמצעות הפקודה KBSCAN, או EKBSCAN. שים לב לכך שלוח מקשים מורחב קיים רק במחשבי AT ומעלה.
התכנית מחכה ללחיצה על מקשים. כל לחיצה על מקש תגרום להצגת קוד הסריקה וערך ASCII של המקש. לחיצה על מקש Esc תסיים את התכנית.
ראה פרק 4 סעיף **4.2.7.2**

KBFLAG.PAS - הצגת מצב הדגלים במקלדת הרגילה

התכנית מציגה באופן קבוע את מצב הדגלים בלוח המקשים.
הפעלת התכנית באמצעות הפקודה KBFLAG.
לחיצה על אחד ממקשי הדגלים (Shift, Ctrl, Alt, Caps Lock, Insert, Num Lock), תשנה את מצבו של הדגל על גבי המסך. לחיצה על מקש שאינו מקש דגלים תסיים את התכנית.
ראה פרק 4 סעיף **4.2.7.1**

KBSPEED.PAS - שינוי מהירות ההדפסה (Typematic rate)

התכנית מאפשרת לשנות את מהירות ההקלדה במקלדת מורחבת במחשבי AT ומעלה.
הפעלת התכנית באמצעות הפקודה KBSPEED.
תנועה בעזרת חצים ימינה ושמאלה. הצד השמאלי של הסקלה מציין מהירות גבוהה. מקש Esc יסיים את התכנית.
ראה פרק 4 סעיף **4.2.4**

MOUSEDEM.PAS - הדגמת פעולת עכבר

לפני הרצת התכנית יש להטעין תכנית הפעלה לעכבר (Mouse Driver) בהתאם לסוג העכבר הנמצא ברשות המשתמש (תכנית זו, למשל MOUSE.COM מסופקת עם העכבר בעת הקניה שלו).
הפעלת התכנית באמצעות הפקודה MOUSEDEM.
התכנית מציירת שני ריבועי בחירה על גבי המסך. לחיצה על אחד הלחיצים של העכבר, כאשר העכבר בתוך אחד הריבועים, תגרום לציור של ריבועים או מעגלים. לחיצה על מקש כלשהו במקלדת תסיים את התכנית.
בתכנית מודגם השימוש במספר שגרות שירות לעכבר. נמצאות בה שגרות שירות נוספות שלא משתמשים בהן כאן (ראה גם תכנית דוגמה PAINT).
ראה פרק 4 סעיף **4.4.5.2**

PAINT.PAS - הדגמה של הפעלת מודול עכבר (ראה תכנית MOUSEDEM)

להפעלת התכנית נחוצים שלושה קבצים: PAINT.EXE, MOUSE.COM, *.BGI.
הקובץ MOUSE.COM, או מקביל לו, מסופק בעת הקניה של העכבר.
הוראות הפעלה לתכנית:
א. הקש את הפקודה MOUSE (למשל, לטעינת תכנית ההפעלה של העכבר).
ב. הקש את הפקודה PAINT.

<u>הוראות התכנית:</u>

1. הפעלת העכבר מאפשרת את הפעולות הבאות:

א. בחירת צבעים ועובי קו: הבא את העכבר למסגרת הימנית. באמצעות לחיצה על אחד מלחיצי העכבר, בחר את מסגרת הצבע או הקו.
ב. ציור נקודות: לחיצה על כפתור ימני בעכבר תצייר נקודה על המסך בצבע שנבחר. כדי לצייר מספר נקודות, לחץ באופן רצוף והזז את העכבר לכיוון הרצוי.
ג. ציור קווים: לחיצה רצופה על כפתור שמאלי של העכבר, הזזת העכבר למקום הרצוי ושחרור הכפתור.
ד. תנועה: הזזת העכבר למקום הרצוי.

2. הפעלת המקלדת מאפשרת את הפעולות הבאות:

המקש	הפעולה
F1	שינוי עובי קו.
0-9	שינוי צבע מ-0 ועד 9. הצבע 0 הוא שחור ומשמש למחיקה.
M	תחילת סימון בלוק לשמירה על גבי הדיסק/דיסקט. לחיצה על כפתור תסמן גמר בלוק. התכנית מבקשת <File Name>, יש להקיש שם קובץ והתמונה תישמר בקובץ <File Name>.GRF. גודל התמונה שניתן לשמור תלוי ברזולוציה של המסך. אם התמונה גדולה מדי יישמע צפצוף. לביטול שמירת הקובץ לחץ על מקש Esc.
L	טעינת קובץ. הקובץ יטען במקום הסמן. יש להקיש את שם הקובץ. אם הקובץ איננו קיים יישמע צפצוף. ביטול שמירה באמצעות לחיצה על מקש Esc.
חיצים	תזוזה על גבי המסך. אם הופעל מקש D להורדת העט, התנועה תגרום לציור של נקודות בצבע שנבחר על גבי המסך.
D	"הורדת" העט. כל תנועה באמצעות המקשים תגרור ציור על גבי המסך.
U	"הרמת" העט. תנועה באמצעות המקלדת לא תגרום לסימון על המסך.
Esc	יציאה.
F2	שינוי של גודל הצעד בתנועת המקשים. כשהצעד גדול אין ציור.

PARK.PAS - נעילת הראשים הקוראים של הדיסק

התכנית מזיזה את הראשים הקוראים של הדיסק C לסוף המשטח המגנטי ומאפשרת לטלטל את המחשב ללא חשש לפגיעה של ראש הקריאה במשטח.
הפעלת התכנית באמצעות הפקודה PARK.
התכנית תפעל רק על מחשבים בהם קיים BIOS חדיש (מ-1986 והלאה).
ראה פרק 5 סעיף 5.5

PRBDEMO.PAS - הדפסת קובצי Text באמצעות פקודות BIOS

התכנית שולחת קובצי Text למדפסת המקבילית. שליחת המידע מתבצעת באמצעות פסיקת השירות 17Hex של BIOS.
הפעלת התכנית באמצעות הפקודה <File-Names> PRBDEMO.
כאשר <File-Names> מפרט את שמות הקבצים שיש להדפיס. ניתן להשתמש בתווי פרא (Wildcards: ?,*).
לדוגמה, הפקודה "PRBDEMO *.PAS *.LST" תדפיס את כל הקבצים המסתיימים ב-PAS, או קבצים המסתיימים ב-LST.
ראה פרק 4 סעיף 4.3.5

PRDDEMO.PAS - הדפסת קובצי Text בצורה ישירה

התכנית שולחת קובצי Text למדפסת המקבילית. שליחת הנתונים מתבצעת באמצעות פקודות ישירות למתאם המדפסת.
הפעלת התכנית באמצעות הפקודה <File-Names> PRDDEMO.
כאשר <File-Names> מפרט את שמות הקבצים שיש להדפיס. ניתן להשתמש בתווי פרא (Wildcards: ?,*).
לדוגמה, הפקודה "PRDDEMO *.PAS *.LST" תדפיס את כל הקבצים המסתיימים ב-PAS, או קבצים המסתיימים ב-LST.
ראה פרק 4 סעיף 4.3.4

PRGDEMO.PAS - הדפסה גרפית למדפסת המקבילית

התכנית מציירת על הדף תשעה מעוינים, זה בתוך זה, באמצעות מדפסת תואמת לתקן של Epson. התכנית מדגימה שליטה בגרפיקה של מדפסות.
הפעלת התכנית באמצעות הפקודה PRGDEMO.
שגרה EscGraphMode מעבירה את המדפסת למצב גרפי, כאשר במשתנה NumOfByteForGraphPrint נמצא את מספר התווים.
ראה פרק 4 סעיף 4.3.3

RSBDEMO.PAS - תכנית תקשורת טורית אסינכרונית באמצעות BIOS

התכנית מאפשרת תקשורת דו-כיוונית פשוטה באמצעות מתאם התקשורת האסינכרונית. היא משתמשת לשם כך בתכניות שירות של BIOS (פסיקת שירות 14Hex).
הפעלת התכנית באמצעות הפקודה RSBDEMO.

כל לחיצה על מקש תשלח את ערכו למתאם, וכל בית שיישלח מצד המחשב יוצג על גבי המסך.

ניתן ליצור התקן לבדיקת תקינות של הערוץ הטורי ובאמצעות התכנית לוודא תקינותו. כדי לבנות את התקן הבדיקה, יש להתקין את החיבורים הבאים בתקע של המתאם הטורי 25pin (לא לחבר את החיבורים ישירות לכרטיס!):

חבר פין 2 לפין 3 בתקע (שידור לקליטה).
חבר פין 4 לפין 5 ו-8 בתקע (הדקי CTS, RTS).
חבר פין 6 לפין 20 בתקע (הדקי DTR, DSR).

בבדיקה עם התקן הבדיקה, כל תו שיישלח לערוץ התקשורת הטורי יוצג גם על גבי המסך, מכיון שערוץ השידור מחובר בעניבה לערוץ הקליטה.

ראה פרק 4 סעיף 4.4.7

RSDDEMO.PAS - תכנית תקשורת טורית אסינכרונית בצורה ישירה

התכנית מאפשרת תקשורת דו-כיוונית פשוטה באמצעות מתאם התקשורת הטורית האסינכרונית. התכנית משתמשת בפסיקות קליטה/שידור לצורך בתקשורת ושולחת את הנתונים בצורה ישירה למתאם התקשורת.

הפעלת התכנית באמצעות הפקודה RSDDEMO.

כל לחיצה על מקש תשלח את ערך המקש למתאם, וכל בית שיישלח מצד שני יוצג על גבי המסך.

ניתן ליצור התקן לבדיקת תקינות של הערוץ הטורי ובאמצעות התכנית לוודא תקינותו. כדי לבנות את התקן הבדיקה, יש להתקין את החיבורים הבאים בתקע של המתאם הטורי 25pin (לא לחבר את החיבורים ישירות לכרטיס!):

חבר פין 2 לפין 3 בתקע (שידור לקליטה).

בבדיקה עם התקן הבדיקה, כל תו שיישלח לערוץ התקשורת הטורי יוצג גם על גבי המסך, מכיון שערוץ השידור מחובר בעניבה לערוץ הקליטה.

בתקשורת הישירה אין צורך בקצרים על הדקי DSR, DTR, CTS, RTS.

ראה פרק 4 סעיף 4.4.4

SCRBDEMO.PAS - כתיבה למסך

תכנית זו משלבת שלושה נושאים:

א. שימוש בשגרות של ROM BIOS לשליטה במסך.
ב. שימוש בפסיקת שעון 1CHex כדי לעורר שגרות.
ג. הכנסת תכנית TSR.

הפעלת התכנית באמצעות הפקודה SCRBDEMO.

נושא א' (מוסבר בפרק 4 סעיף 4.1.8) מודגם על ידי שתי שגרות:
Bioswritestr כותבת מחרוזת תווים למסך, בדף P במקום המוגדר על ידי הקואורדינטות X,Y.
Biosswap מחליפה דף הצגה אקטיבי המוגדר על ידי P.
בצורה זהה ניתן לבנות לפי ההנחיות שבסעיף 4.1.8 שגרות לשליטה מלאה על המסך ללא שימוש ב-DOS. הן תבצענה את המלאכה בצורה מהירה יותר מאשר כתיבה דרך DOS.

נושא ב' מודגם בפרוצדורת TICKER. ראה תכנית דוגמה CLOCK.
זוהי שגרת פסיקת חומרה. ראה הסברים בעניין זה.

נושא ג' - תכנית TSR.
ב-Turbo Pascal הוספה הפקודה KEEP שמפעילה את הפסיקה 27Hex. היא מסיימת את התכנית, אבל משאירה את התכנית בזיכרון. מכאן, ששגרות הפסיקה בתכנית ימשיכו לפעול (ברקע) גם לאחר סיום התכנית.

תכנית זו תפעל רק במסכים צבעוניים, מאחר ובמסך הרקולס אין תמיכה בדפים.

SCRDDEMO.PAS - כתיבה ישירה למסך

התכנית כותבת תווים אקראים עם תכונות אקראיות במקומות אקראיים על פני המסך. השורה התחתונה במסך (מספר 25) נשמרת ריקה, כך שהכתיבה הרגילה במקלדת מוצגת בשורה זו. שים לב, שמילוי המסך מתבצע ללא פגיעה בסמן שנמצא וזז בשורה זו כתוצאה מפעולות ההקלדה.

הפעלת התכנית באמצעות הפקודה SCRDDEMO.

אין אנו ממליצים למשתמש להשתמש בטכניקה לכתיבה ישירה, אלא בטכניקה של BIOS בלבד (ראה תכנית SCRBDEMO).

SCRGDEMO.PAS - הדגמה של כתיבה גרפית ישירה למתאם המסך VGA, EGA

התכנית מציירת קו אלכסוני על גבי המסך.
הפעלת התכנית באמצעות הפקודה SCRGDEMO.
סיום התכנית על ידי לחיצה על מקש Enter.
התכנית מדגימה 5 פעולות במתאם המסך:
1. מעבר ל-Mode גרפי ב-Pascal.
2. ציור קו אדום ב-Pascal.
3. ציור קווים צבעוניים על ידי שימוש ב-BIOS.
4. ציור קווים צבעוניים על ידי שימוש בכתיבה ישירה למרחב הזיכרון של הכרטיס.
5. שינוי טבלת הצבעים בכניסה הלבנה (15) על ידי שימוש ב-Pascal.

ראה פרק 4 סעיף 4.1.5.3

SETALARM.PAS - הפעלת שעון זמן אמת וכיוון שעון עצר

התכנית נטענת כתכנית TSR בזיכרון המחשב.
התכנית משמיעה ציפצוף ורושמת הודעה על המסך בשעה שכוונה בשעון העצר.
התכנית מאפשרת לכוון את שעון העצר לשעה הרצויה.
הפעלת התכנית באמצעות הפקודה "SETALARM time message", כאשר:
time - שעת שעון העצר. אם לדוגמה קבענו 1230, הציפצוף ישמע בשעה זו.
message - הודעה שתוצג בשורה הראשונה במסך.
בכל פעם שתופעל התכנית, שעת שעון העצר תכוון מחדש על פי time.

SOUND.PAS - השמעת ציפצוף ברמקול

התכנית משמיעה ציפצוף בתדרים 50-1000Hz (בקפיצות של 50Hz).
הציפצוף נשלח בשלב ראשון באמצעות כתיבה ישירה לרכיב 8254 המחובר לרמקול, ובשלב שני – באמצעות פקודות הכלולות בשפת פסקל.
הפעלת התכנית באמצעות הפקודה SOUND.

תכנית TREEDEMO.PAS – הצגת מבנה הספריות

התכנית מציגה את מבנה הספריות של הדיסק/דיסקט שבו נמצא המשתמש.
התכנית עוברת על פני כל הספריות ותת-הספריות ומציגה את מבנה העץ בדומה לתכנית TREE של DOS.
הפעלת התכנית באמצעות הפקודה TREEDEMO.
ראה פרק 5 סעיף 5.4.3

VIRDEMO - הדגמה של תכנית וירוס

הפעלת התכנית באמצעות הפקודה VIRDEMO.
זהירות!!! הפעלה של התכנית מדביקה בוירוס את כל הדיסקים/דיסקטים במערכת. עם זאת, תכנית הוירוס "ידידותית", אינה מזיקה ואינה "מסתתרת". על כן תוכל להסיר אותה בקלות על פי ההנחיות, גם ללא שימוש בתכניות אנטי-וירוס (אשר מצאנו שהן אינן מגלות אותו).

לאחר שהדיסקים הודבקו, תכנית VIRDEMO תפעיל את עצמה בכל "הדלקה" של המחשב. כדי לנקות את המערכת יש לפעול בדייקנות על פי ההנחיות המופיעות על גבי המסך: יש למחוק את כל קובצי VIRDEMO.EXE ואת השורה הראשונה בכל קובצי AUTOEXEC.BAT שבכל הדיסקים/דיסקטים שהופעלו במחשב בעת הפעלת התכנית לראשונה.
השגרה Virus-Sabotage פועלת בזמן אקראי ומציגה הודעה על גבי המסך.
השגרה Virus-Spread פועלת רק בהרצה של התכנית ומדביקה את הדיסקטים.
בתכנית אין שגרת הסתתרות על מנת לאפשר למשתמש להוריד את הוירוס בקלות. כמו כן בחרנו שלא ליצור תכנית התפשטות הפועלת כל כמה דקות, אלא בחרנו בהתפשטות המתבצעת רק פעם אחת בתחילת התכנית, על מנת לא לספק כלי "חבלה". באמצעות תכנית זו רצינו:
א. להדגים את עקרונות הפעולה של וירוסים ולא לגרום לחבלה!
ב. להציג מצב שבו אין תכנית אנטי-וירוס שמגלה את וירוס הדוגמה.

השתמשנו בטכניקה מסוימת ליצירת הוירוס, כדי להדגים שאין אפשרות לחסום מערכת מחשב לחלוטין בפני וירוס ושניתן ליצור וירוסים שיוכלו לעקוף תכניות אנטי-וירוס קיימות!

תרשימים וטבלאות

תרשימים

טבלאות

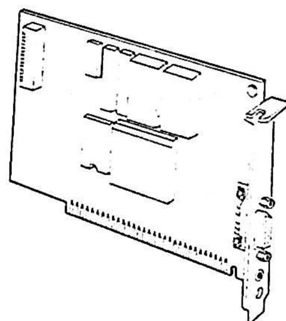
אינדקס

האינדקס מחולק לשני מדורים: האחד – בעברית והשני – באנגלית.

האינדקס נותן את מראי המקומות העיקריים של הדיון בנושא, או במונח.

המדור בעברית כולל בדרך כלל גם את המונחים המקבילים באנגלית ועל כן הוא יכול לשמש כמו מילון פשוט, אשר מתייחס למונחים שבספר.

אינדקס עברי



אינדקס אנגלי

D

E

ביבליוגרפיה

רשימת הספרים עבור משתמשי PC מקיפה תחומים רבים. בתחום שבו עוסק הספר אפשר להיעזר בספרות העניפה של היצרנים ושל מפתחי רכיבים ומערכות עזר. בחרנו לציין מספר פריטים.

IBM Personal Computer AT Technical Reference
IBM Corporation

IBM Personal Computer XT Technical Reference
IBM Corporation

IBM Personal System/2 Model 50 and 60 Technical Reference
IBM Corporation, 1987

Microsoft MS-DOS Getting Started and User's Guide and Reference
Version 5 / Microsoft Corporation, 1991

Norton Commander / Norton Utilities
Peter Norton, 1990

PC Tools Delux Version 6
Central Point Software, 1990

386 SX Microprocessor
Intel Corporation, 1989

Terry Dettmann / Jim Kyle
DOS Programmer's Reference (2nd Ed)
Que Corporation, 1989

Mark Minasi
The Hard Disk Survival Guide
Sybex, 1991

Mark Minasi
The Complete PC Upgrade and Maintenance Guide
Sybex, 1991

Alan Simpson
Best Book of MS-DOS 5
SAMS, 1991

Ed Teja and Laura Jones
IBM PC and PS/2 Graphics Handbook
Microtrend Books, 1990

John Woram
PC Configuration Handbook
Bantam, 1991

ד"ר דניאל גלין
פיתוח מערכות מידע אישיות / הוד-עמי, 1991

זהר עמיהוד
DOS 5 – המדריך השימושי / הוד-עמי, 1991

ג'פרי י. קרנץ, אנה מ. מיזל, רוברט ל. וויליאמס
OS/2 – מאפיינים, פונקציות ויישומים / הוד-עמי, 1989



הוצאת הוד-עמי
לספרי מחשבים

ת.ד. 6108 הרצליה, 46160
טל': 052-541207, פקס: 052-571582

(מוזגשים ספרים שפורסמו בשנים 1990, 1991)

DOS 5 — מדריך שימושי. זהר עמיהוד
עיקרי ההפעלה והשימוש במחשב האישי ובדיסק הקשיח. מתאים גם לגירסאות קודמות.

פיתוח מערכות מידע אישיות. ד"ר ד' גלין.
תכנון יישומים למחשב האישי בבית העסק או בבית. דוגמאות רבות ואירוע תכנון מפורט.

אבטחת מידע במערכות ממוחשבות. ו' ליין.
עקרונות ושיטות הגנה. אבטחה פיסית, בקרה, הצפנה, מערכות תקשורת ומערכות מקוונות, מחשבים אישיים. חוק צנעת הפרט וחוק המחשבים בישראל.

טורבו-פסקל - תכנות מבני למתחילים ומתקדמים (גירסאות 5.5 ו-6) נרוני א' אנגלנרו.
כולל: Tools, editor, תכנות מבני, מבני נתונים, רקורסיה, קבוצות, תצוגת מסך ועוד.

פסקל - שפה לתכנות מבני. נרוני א' אנגלנרו
רמה תיכונית ואוניברסיטאית, מבני נתונים, רקורסיה, קבוצות, דוגמאות ותרגילים.

ספר התשובות לפסקל. אמ"ר קנלן.
פתרונות לשאלות ותרגילי תכנות בספר "טורבו-פסקל" ו"פסקל".

המדריך השלם לשפת C. ע' רש ו-מ' ליכטמן.
טכניקות מתקדמות לתכנות ב- Turbo-C, קדם-מהדר, שילוב עם UNIX. כולל דיסקט תרגול.

ללמוד U N I X. א' כילאווקים.
למתחיל ולמתקדם - כלי תכנות ושירות ושימוש ב-C. לתחזוקה - גיבוי וטיפול בתקלות.

קובול 85 מבני - למחשבים אישיים, מחשבי VAX ומחשבים מרכזיים. נרוני א' אנגלנרו.
הדרכה למתכנת, תכנות מבני, פונקציות חדשות, הדרכה לתכנות יעיל. תוכניות מוסברות.

לוטוס 123 למתקדמים - שפת המקרו. א' אולמן.
לימוד שימוש במקרו לבניית יישומים. דוגמאות. דיסקט תרגול.

מקרו למתקדמים בלוטוס - Meta-Macro ו-Intra-Macro (יישומים מתקדמים). ר' אברהמי.
יישומי מקרו מתקדמים, שליטה בגיליון ובמאגרי הנתונים, תפריטים, אבטחת מידע.

מחוללי יישומים dBASE III Plus ו-dBXL למשתמש בעברית. א' רון. >>מהדורה מעודכנת<<
למתחיל - תפריטים ומסכים. למתקדם - תכנות בדור הרביעי, איחזור מידע, דוחות ועוד.

הכנה לבגרות במחשבים - מבוא למחשבים, תכנות, ענ"א. ל' רוקח.
שאלונים במתכונת בגרות עם פתרונות. בסיסי מספרים, יישומים מדעיים ומערכות ספרתיות.

התלמיד והמחשב - 80 פתרונות. מ' הומי.
פתרון בעיות, חידות ושעשועים בבייסיק ופסקל.

OS/2 - מאפיינים, פונקציות ויישומים. קרנץ, מיזל, וויליאמס.
הכרת מערכת ההפעלה והשוואה ל-DOS. הסבר המעבדים 286/386. Presentation Manager, Database Manager.

פרולוג שפת בינה מלאכותית. ד"ר נ' צירולניקוב.
לוגיקה פסוקית, תכנות לוגי, בניית מערכת מומחה. דוגמאות.

LOGO - תכנות פונקציונלי. ש' מנולה.
תכנות מבני, פונקציות ופרוצדורות, רקורסיה, רשימות, דוגמאות פתורות ותרגילים.

יסודות בניתוח ותכנון מערכות מידע. עמיהוד ו-קורבל.
מחזור חיים לפיתוח מערכת, הנחיות והמלצות לכיווני פעולה.

הנדסת תוכנה - הלכה ומעשה בפיתוח מערכות. מ' נתניאס.
גישות, מתודולוגיות, טכניקות וכלים מתקדמים. המלצות לעבודה יעילה ודוגמה מפורטת.

ארגון נתונים וניהול קבצים. ר' הייברמן.
מבני נתונים ומאפייני קבצים, תכנון רשומות וניהול קבצים.

בסיסי נתונים - עקרונות, מודלים ויישומים. ר' הייברמן.
הסבר המודלים ומערכות מסחריות, מערכות בסיסי נתונים טבלאיים (Relational).

תקשוב ורשתות. א' סקוב.
רכיבי חומרה ותוכנה, יישומי רשתות רחבות ומקומיות. תקני OSI, תכנון רשת, ביזור.

אר"מ - S.N.A - ארכיטקטורת רשת מערכות תקשורת. נ' שחר.
מרכיבי החומרה והתוכנה ברשת התקשורת.

המחשב האלקטרוני - עיבוד נתונים ותכנות בייסיק. י' האן ו-י' עמיהוד.
חומרה ותוכנה, תקשורת, מערכות הפעלה, ניהול נתונים, מיני ומיקרו מחשבים, בייסיק.

מיקרו-מחשבים וה-6800 - חומרה ותוכנה. א' הראל.
מערכות תקשורת נתונים. א' סקוב.
תכנון ותכנות יישומי מחשבים. י' עמיהוד.

תכניות דוגמא לספר

המחשב האישי למשתמש המקצועי

מחשבי יבמ PC ותואמים

הוצאת הוד - עמי לספרי מחשבים בע"מ
ת.ד. 6108 הרצליה 46160, טל: 052-541207

המחשב האישי למשתמש המקצועי

מיועד למי שיודע ורוצה לשפר, לזה שיודע מעט ורוצה ללמוד ולהתקדם, ולכל מי שרוצה להיות "מקצוען" במחשב האישי (PC) של יבמ ותואמיו ובמערכות PS/2 ו-PS/1 של יבמ.

מה תמצא בספר?

- **שליטה מקצועית ברכיבי המערכת** לניצול יעיל ולשיפור התפוקה.
- **החומרה של ה-PC:** מעבדי משפחת 80X86 של אינטל, הלוח הראשי ורכיביו, ROM BIOS, זיכרונות, כרטיסי הרחבה ותיאום.
- **התקני קלט ופלט:** המקלדת, המדפסת, המסך, תקשורת אסינכרונית ורשתות תקשורת.
- **הדיסק הקשיח והדיסקט:** אחסון נתונים, שחזור ושיפור ביצועים.
- **מערכת ההפעלה:** מבנה פנימי, פסיקות, מיפוי מרחבי הזיכרון ומרחבי קלט/פלט. פקודות יחודיות לניהול, בקרה ותחזוקה. רשימת הפקודות של DOS 5.
- **מבט אל הקרביים** של המערכת באמצעות DEBUG, מבנה קובצי תוכניות וחדירת וירוסים למערכת.

דיסקט תוכניות דוגמה כולל אוסף של תוכניות שירות, אשר מציגות יישומים מגוונים (גם תכנית וירוס ידידותית!) בכלי החומרה והתוכנה ומאפשרות ללומד לתרגל ולהשתמש בהן לתועלתו.

למי מיועד הספר?

- למי שרוצה להכיר את ה"קרביים" של המחשב ואופן פעולתו;
- למי שמתכנן ובונה מערכות משובצות במחשבים אישיים;
- למי שנותן שירותי מיחשוב בארגון וכמובן,
- ספר חובה למשתמש המקצועי, לטכנאי ולנותן שירות ללקוחות.

משה קליג', תואר ראשון (BSC) במדעי המחשב ותואר שני (MA) במינהל עסקים ומערכות מידע מהאוניברסיטה העברית בירושלים. מרצה במדעי המחשב ומנחה פרוייקטים בחומרה ותוכנה במחשבים אישיים בחברת בזק.

עידו שרון, מהנדס אלקטרוניקה (BSC) מהטכניון בחיפה, ראש צוות פיתוח לפרוייקטים בחומרה ותוכנה של מחשבים אישיים בחברת בזק.

מסת"ב-1-017-361-965-ISBN